

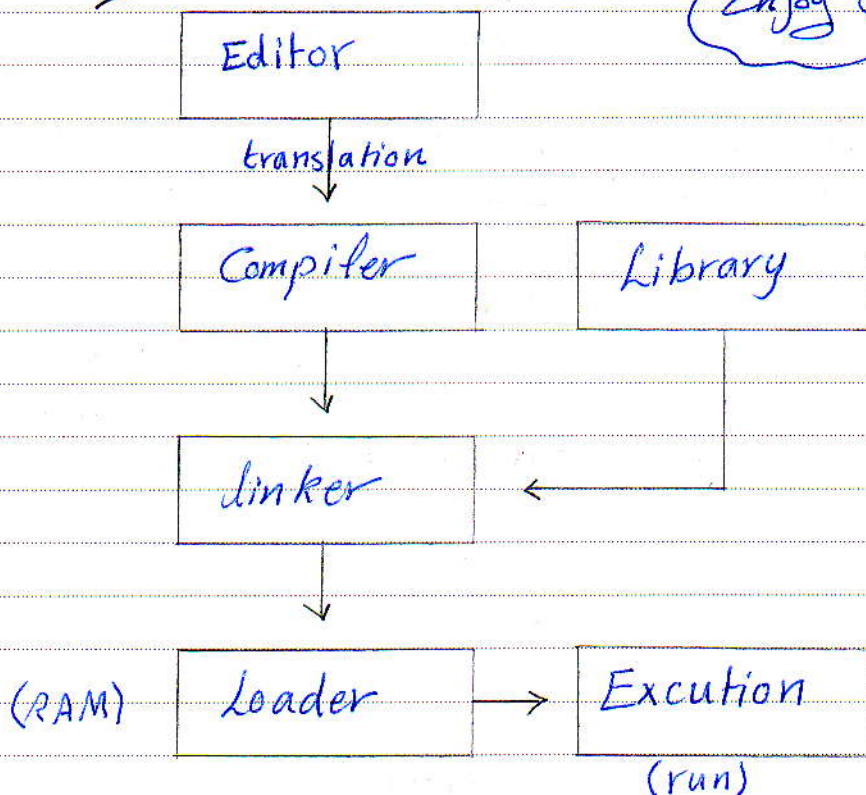
* Chapter One :-

* Probleming language :-

- 1) Machine language. 0s1
 - 2) Assembly language. Add
 - 3) H.L.L. (C++).
-] L.L.L

* C++ environment :-

Enjoy C++



Subject:

/ /

* Problem $\rightarrow$ Analysis $\rightarrow$ Alg $\rightarrow$ Flowchart
code. $\leftarrow$

* Alg :- 1) start 2) Input 3) process
4) output 5) End.

* Example) Find the area of a rectangle. by Alg.

Alg :- 1) start
2) Read L, W
3) Area = $L \times W$
4) Print Area
5) End.

* Flowchart symbols :-



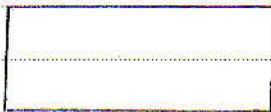
start / End



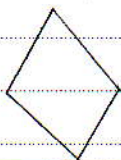
connection



Input / output



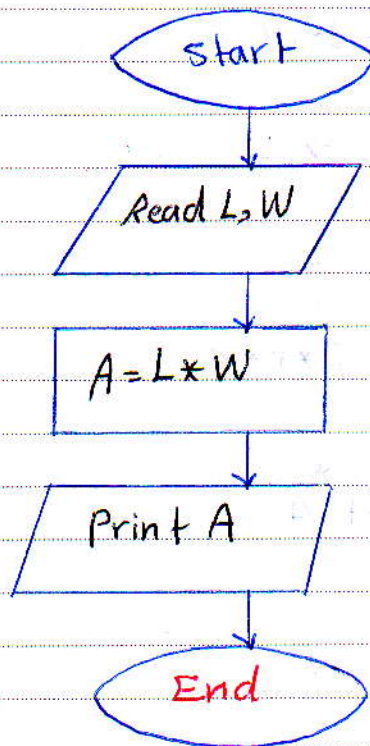
process



condition

Subject:

* Example) Draw a flowchart to find the area of a rectangle.

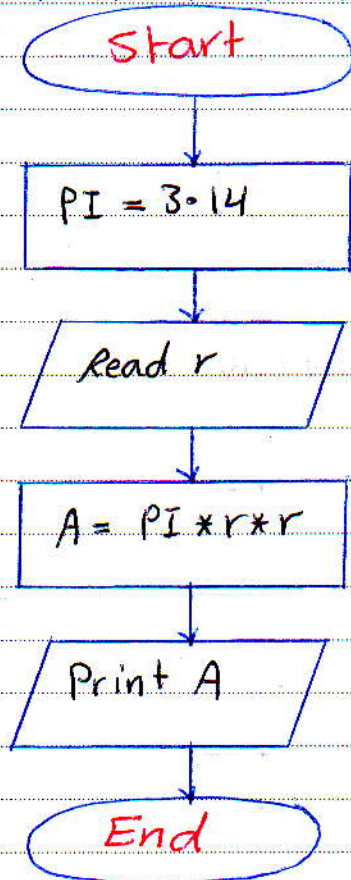


* Example) Draw a flowchart to find the area of a circle.

$$A = \pi r^2$$

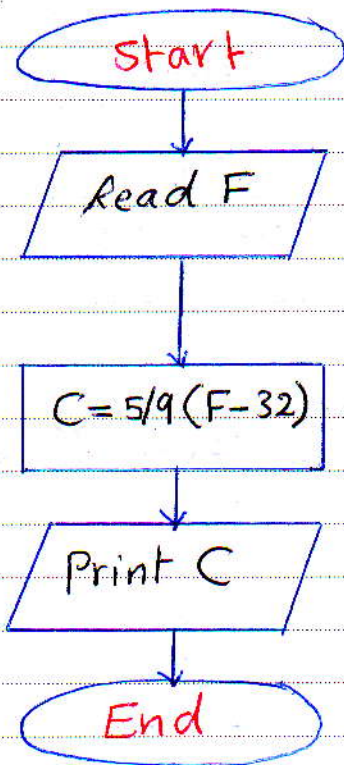
π ← ثابت يظل تعريفه في الحفظ في بداية.

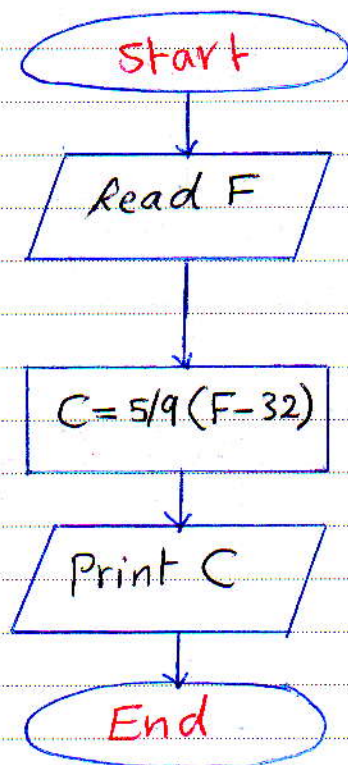
Subject:



* example) Draw a flowchart that convert a temperature in degree fahrenheit to degree celsius.

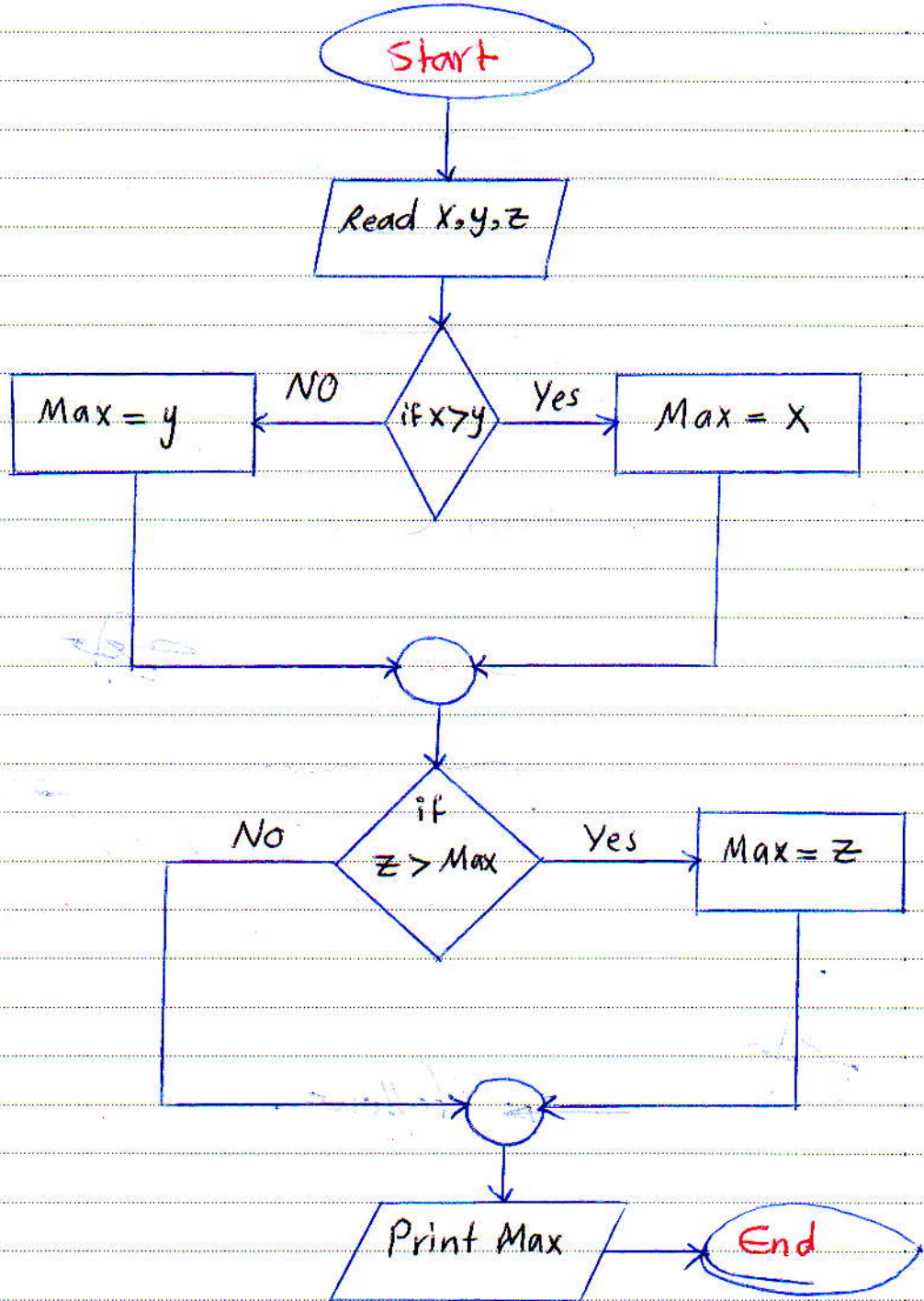
where $\Rightarrow C = \frac{5}{9} (F - 32)$.

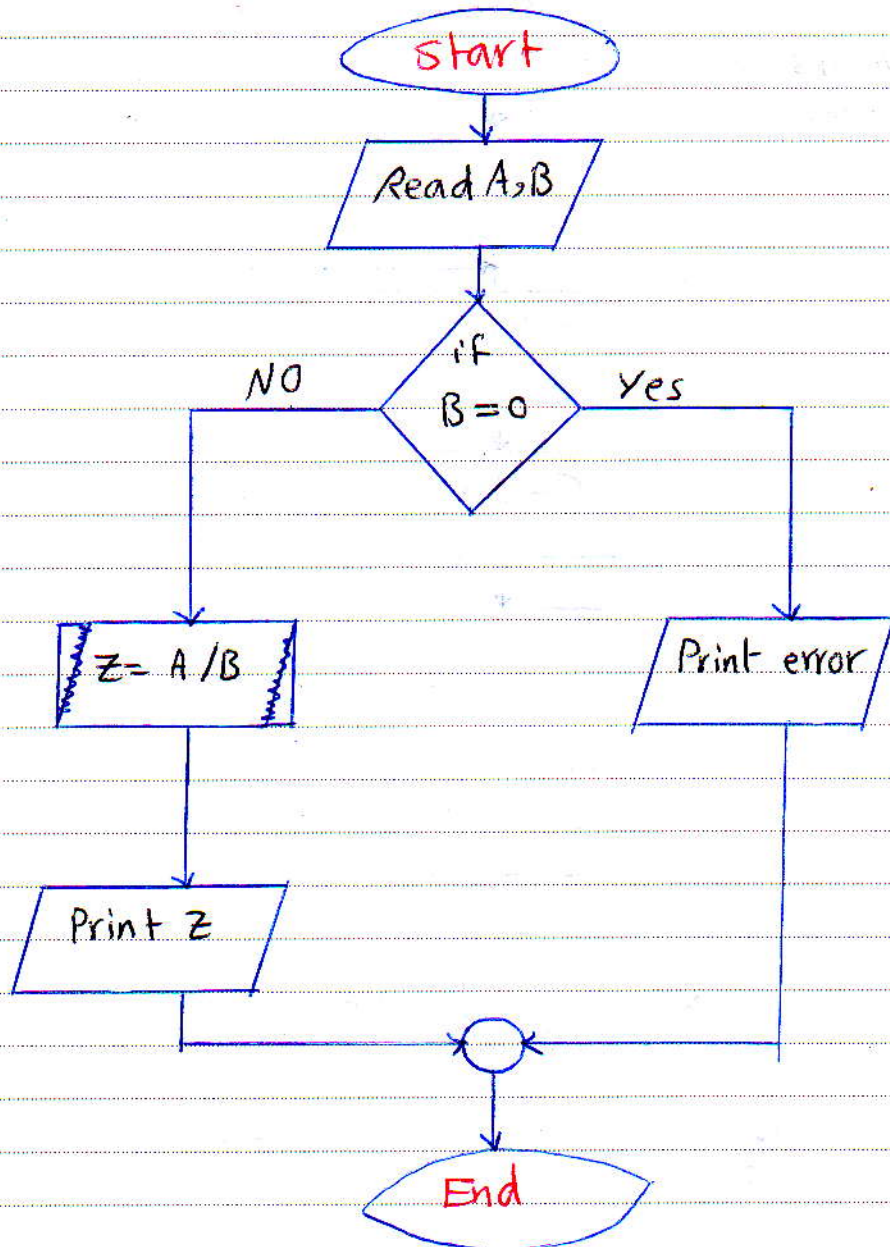




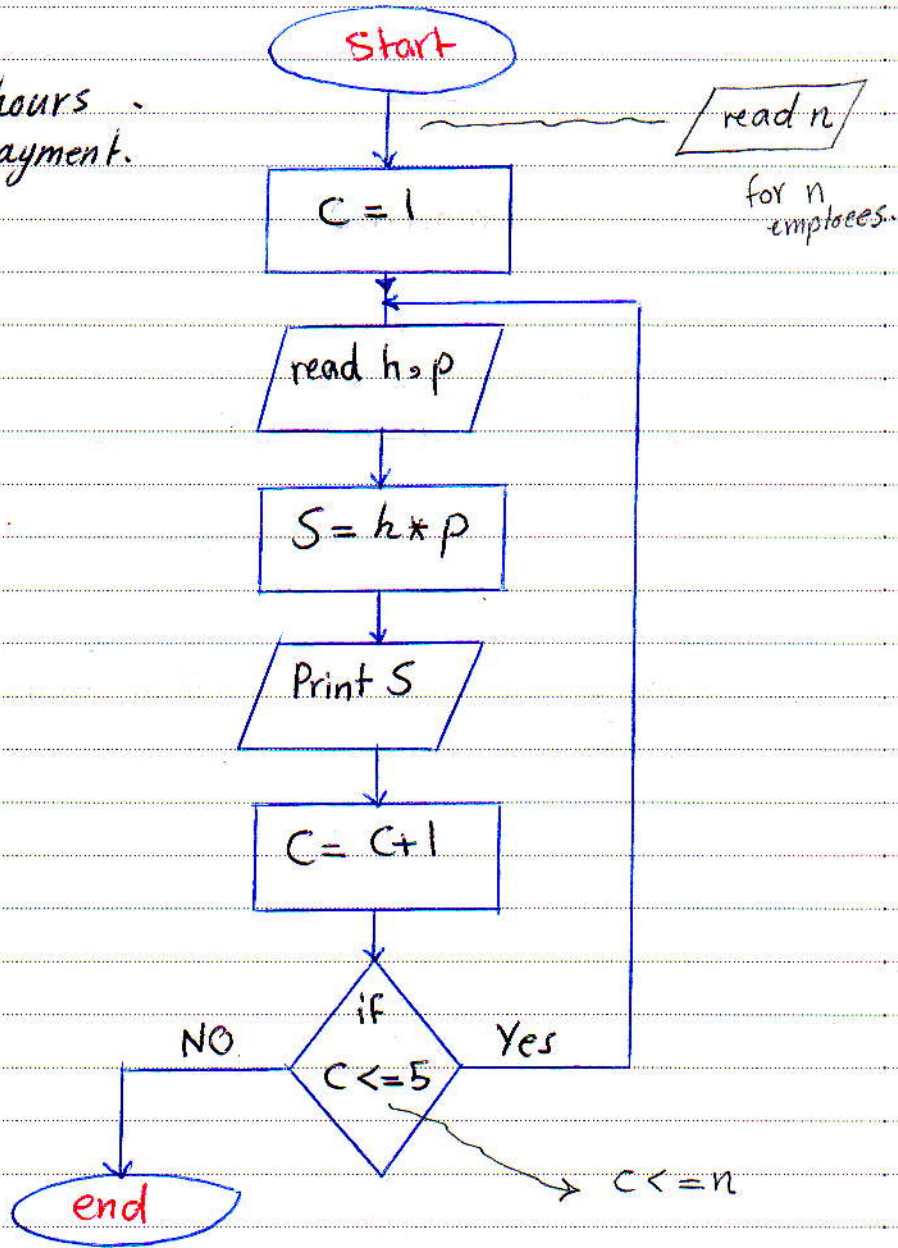
* Example) Draw a flowchart to find the maximum of 3 integer numbers.

→ follow.





h : hours .
 P : payment.



Start

read n

C = 1, min = 100

read g

if $g < \min$

min = min

min = g

C = C + 1

if $C \leq n$

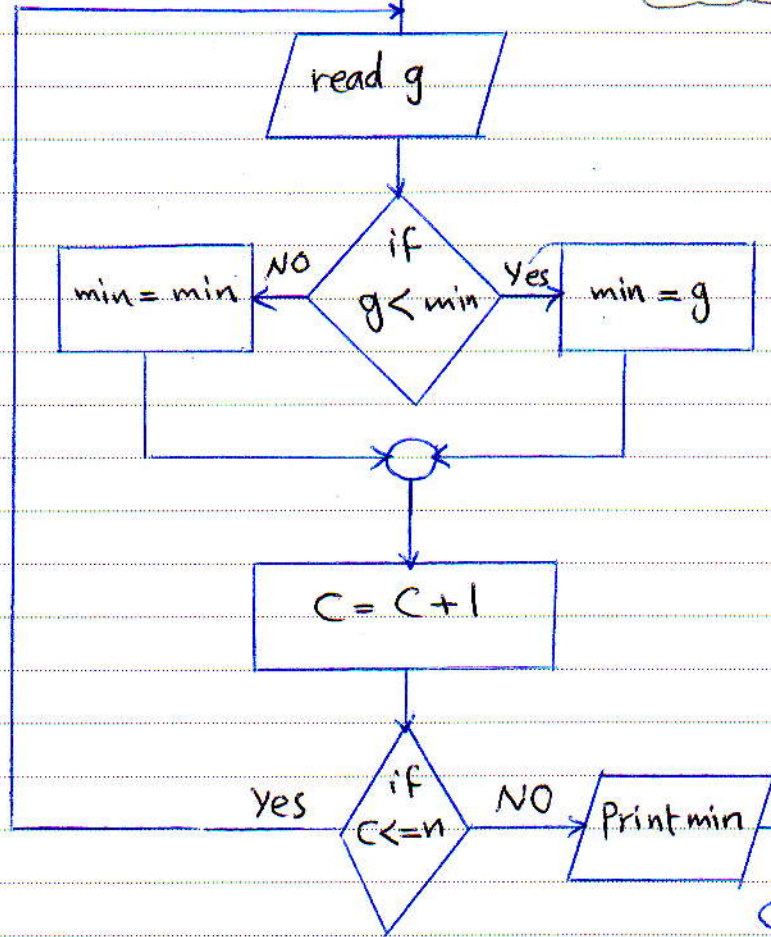
Print min

end

پایان

تغییرات اوله قیلا گ
علیلا min
ثم مقایسه، اطلاعات
الآخری بها

عملیات الترتیب به ترتیب
عبارت به ترتیب، لا یستطیع



Chapter 2

Subject:

* The Basic of C++ language :-

▣ Special symbols :-

- 1) Arithmetic symbols $\rightarrow (+, -, *, /, \%)$
- 2) Logic symbols and connectors $\rightarrow (||, \&, !)$
 $\rightarrow (<, >, >=, <=, ==, !=)$ and or not
Is equal not equal
- 3) Punctuation symbols $\rightarrow (, ; , / , \backslash , \dots)$
جملة أو حرف منطق

* cout << 'a' ; ✓
* cout << "a" ; ✓
* cout << "welcome" ; ✓
* cout << 'welcome' ; ✗

▣ Word symbols : use small letters only. (sensitive language.)

كلمات محروقة C++

⚠ Key words (reserved word) :-

for, while, do, if, else, switch, case, int,
float, char, true, false, static-cast.

عشري
رمز
عنصر

2 Identifiers

Name of variables.

" " constant.

" " Functions.

* Rules of Identifiers :-

- 1) It consists of characters (a-z, A-Z) and digits (0 ... 9) and underscore. (_)
- 2) It must begin with character.
- 3) It must not contain any special symbols.

* Example) Which of the following is valid identifiers :-

✓ counter1

X Employee salary

✓ X₁✓ H₂O

✓ Employee_salary

_room ✓

Hello! X

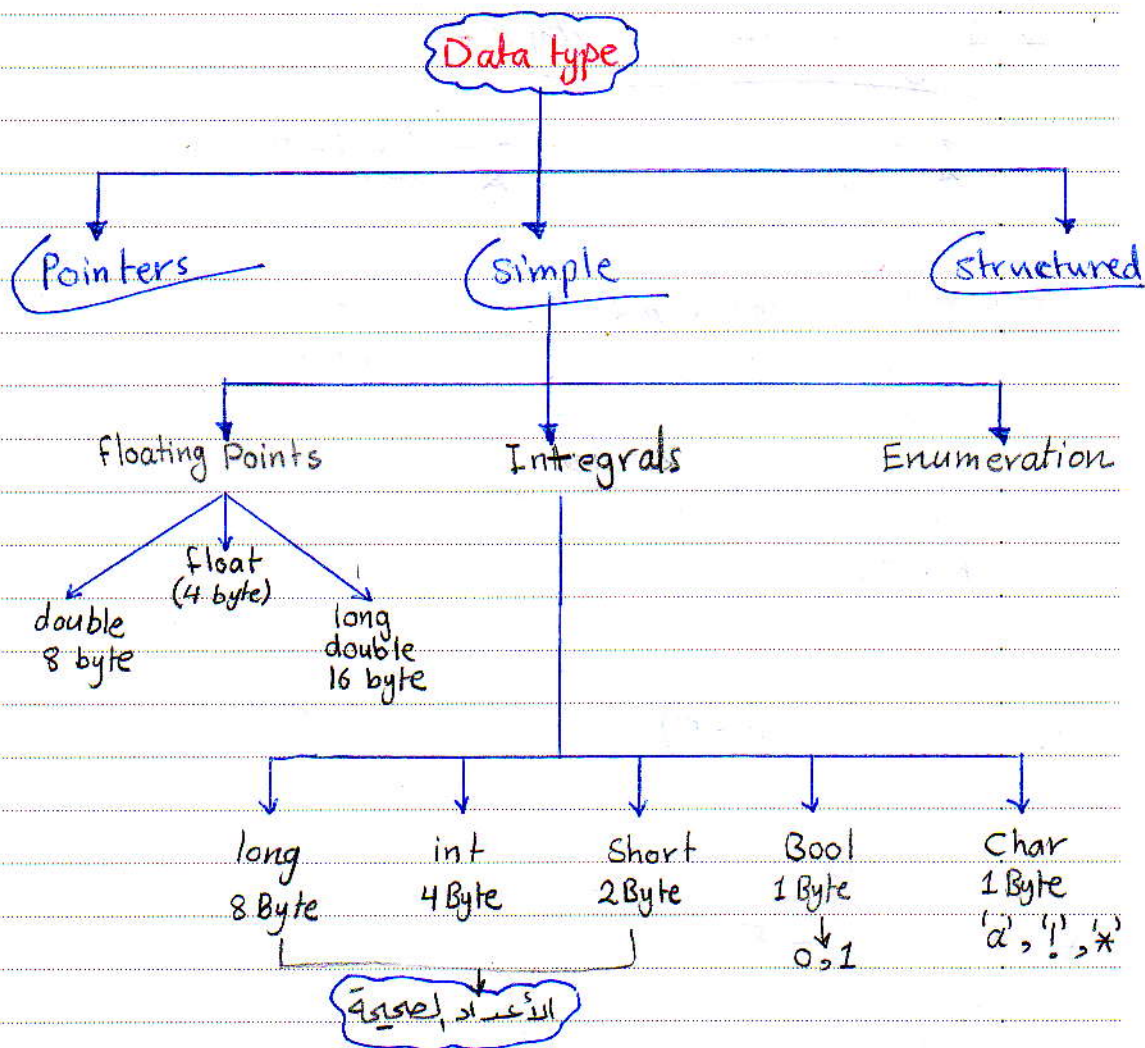
one + two X

2nd X

-2room ✓

Subject: / /

* Data Type :-



$X = 3 < 2 ;$
0 $\leftarrow X = \text{false} ;$
 $X \rightarrow \text{bool}$

True $\rightarrow 1$
False $\rightarrow 0$

* Arithmetic operators & precedence :

* Assignment operator (=)

$$a = 2;$$

$$a = b + d - e$$

* example

int a, b;

or

int a;
int b;

① $a = 10;$

$$\frac{a}{10}$$

$$\frac{b}{??}$$

$$b = 4;$$

$$10$$

$$4$$

$$a = b;$$

$$4$$

$$4$$

$$b = 7;$$

$$4$$

$$7$$

$$b = a;$$

$$(4)$$

$$(4)$$

② $a = 2 + (b = 5);$

$$a = 2 + 5$$

$$a = 7$$



③ $a = b = c = 5$

$$c = 5$$

$$b = 5$$

$$a = 5$$

* Arithmetic operators :-

(+)	add	} Integers + Floating point
(-)	sub	
(*)	Mult	
(/)	Div	
(%)	mod	} Integers only.

* There is two types of operators :-

- 1) Binary operators. $(x+y) / (x-y)$.
- 2) Unary operators. $+5, -4$

Unary $\leftarrow -5 + X$ Binary.

Expression	Result
$34 - 20$	14
$2 + 5$	7
$5 / 2$	2
$14 / 7$	2
$34 \% 5$	4
$-34 \% 5$	-4
$34 \% -5$	4
$-34 \% -5$	-4
$4 \% 6$	4
$5.0 / 2.0$	2.5
$5.0 / 2 \rightarrow 5.0 / 2.0$	2.5
$5 / 2.0 \rightarrow 5.0 / 2.0$	2.5

amr alil 5.0
fawad, 5.0

* precedence %

الأولويات

- 1) ()
- 2) * / %
- 3) + -
- 4) =

* Example) Solve for x.

$$X = 3 * 7 - 6 + 2 * 5 / 4 + 6$$

$$X = 21 - 6 + 2 * 5 / 4 + 6$$

$$= 21 - 6 + 10 / 4 + 6$$

$$= 21 - 6 + 2 + 6$$

$$X = 23$$

* What is the last Arithmetic operator? $\Rightarrow$ add (+)* = = = = operator? $\Rightarrow$ Assignment (=).

* Expressions

Expressions

Integers
expressionsFloating points
(decimal expressions)mixed
expressions

* examples

$$1) \text{ static_cast} \langle \text{float} \rangle (15) / 2;$$

$$15.0 / 2. = 15.0 / 2.0 = 7.5$$

$$2) \text{ static_cast} \langle \text{float} \rangle (15/2);$$

$$(15/2) = (7) = 7.0$$

$$4) \text{ static_cast} \langle \text{int} \rangle (7.8 + \text{static_cast} \langle \text{double} \rangle (15) / 2);$$

$$\text{static_cast} \langle \text{int} \rangle (7.8 + (15.0) / 2.0);$$

$$\Rightarrow \text{static_cast} \langle \text{int} \rangle (7.8 + 7.5);$$

$$\Rightarrow \text{static_cast} \langle \text{int} \rangle (15.3);$$

$$\Rightarrow 15$$

$$5) \text{ static_cast} \langle \text{int} \rangle (7.8 + \text{static_cast} \langle \text{double} \rangle (15/2));$$

$$\text{static_cast} \langle \text{int} \rangle (7.8 + 7.0);$$

$$\Rightarrow \text{static_cast} \langle \text{int} \rangle (14.8);$$

$$\Rightarrow 14$$

$$78.924 \longrightarrow 7.8924 E_1$$

$$0.18 \longrightarrow 1.8 E_{-1}$$

$$7800.0 \longrightarrow 7.8 E_3$$

$$-1.23 \longrightarrow -1.23 E_0$$

* Structure of a C++ program :-

// This is My first C++ program.

include < iostream.h >

start. [int main ()
{

cout << "welcome to My program\n";

return 0;

end [}

تحتي أن
البرنامج نفذ بشكل
صحيح

* comments

single line

//

/* */

multiple line

/*

..... */

input / output

include < iostream.h >

directive

pre processor.

cmath (math.h)

ctime (time / date)

Subject:

int main () → return 0; ~~g. 1234~~
int void () → return 0; ~~g. 1234~~

* To print on screen we use

cout <<

→ insertion stream.

* cout :-

① cout << "Welcome to my programme\n";
Welcome to my programme.

② cout << "Welcome to";
cout << " my programme\n";

Welcome to my programme

③ cout << "Welcome to"
<< " my programme\n";

Welcome to my programme.

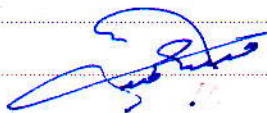
④ cout << "welcome\n to\n my programme\n";

Welcome
to
my programme.

Subject: _____

⑤ `cout << "Welcome\n"`
`<< "to\n"`
`<< "my programme\n";`

Welcome
to
my programme.



⑥ `cout << "Welcome\n";`
`cout << "to\n";`
`cout << "my programme\n";`

Welcome
to
my programme.

* Example)

- ① `cout >> 29/4 ;` → 7
② `cout >> "Hello there" ;` → Hello there
③ `cout >> 12 ;` → 12
④ `cout >> "4+7" ;` → 4+7
⑤ `cout >> 4+7 ;` → 11
⑥ `cout >> "4+7=" << 4+7 ;` → 4+7 = 11

* Escape Sequences :-

$\backslash n$	→	new line
$\backslash t$	→	tab (فواصل)
$\backslash r$	→	carriage return.
$\backslash b$	→	backspace.
Action ← $\backslash "$	→	cout << "Hello\" ; → Hello\

1) cout << "Hello\n there";

Hello
there

2) cout << "Hello\t there";

Hello there.

3) cout << "Hello\r there";

there.

4) cout << "welcome\r to me";

~~to me~~
to meme

5) cout << "Hello\b there";

Hell there

Subject:

6) `cout << "Hello \b there";`

Hello there.

7) `cout << "Hello \b\b there";`

Hel there.

8) `cout << "welcome \r to c++";`

to c++ me.

9) `cout << "/" "Hello there/" "`

"Hello there"

10) `cout << '/' 'a/' "`

'a'

*Cin

`int x, y; int sum;`

`cin >> x;`

`cin >> y;`

`sum = x + y;`

`cout << sum;`

`int a, b, c;`

or

`int a;`

`int b;`

`int c;`

* Initializing variables :-

* `int c;`
`c = 1;`

C++

* `int c = 1;`

* constant declaration :-

`const float PI = 3.14;`

`PI = PI + 3;` "error"

* Increment & Decrement operators :-

`a = a + 1;`

`a = a - 1;`

`++a;`

`a++;`

`--a;`

`a--;`

⊛ `a = 2;`

`x = 1;`

`x = x + (++a);`

`x = x + (a++);`

a x

3 4

4 7

⊛ `A = 3;`

`B = ++A;`

`B = A++;`

A B
4 4

A B
4 3

Subject:

*ex) $a = 5;$ $b = 6;$

⇒

$a = (b++) + 3;$

$c = 2 * a + (++b);$

$b = 2 * (++c) - (a++).$

<u>a</u>	<u>b</u>	<u>c</u>
9	7	??
9	8	26
10	45	27

* Compound assignment Operators :-

- 1) $X = X + 3 ;$ → $X+ = 3;$
- 2) $X = X * 3 ;$ → $X* = 3;$
- 3) $X = X - 3 ;$ → $X- = 3;$
- 4) $X = X / 3 ;$ → $X/ = 3;$
- 5) $X = X \% 3 ;$ → $X\% = 3;$

* example

- 1) $X = 2 * X ;$ → $X* = 2;$
- 2) $X = X + y - 2$ → $X+ = y - 2;$
- 3) $Z = Z * X + 2 * Z$ → $Z* = (X + 2);$
- 4) $y = y / (X + 5)$ → $y/ = (X + 5);$
- 5) $Z = Z * X + 2$ → $Z* = X + 2 / Z;$

Subject:

/ /

*example)

```
int a, b, sum ;  
double c ;  
a = 3 ;  
b = 5 ;  
c = 14.1 ;
```



	<u>a</u>	<u>b</u>	<u>c</u>	<u>sum</u>
sum = a + b + c ;	3	5	14.1	22
c /= a ;	3	5	4.7	22
b += c - a ;	3	6	4.7	22
a * = 2 * b + c ;	50	6	4.7	22



C++

* Chapter Three 3 :-

* Relational Operators :-

$=$	$\longrightarrow$	Is equal.
$!=$	$\longrightarrow$	not equal.
$>$	$\longrightarrow$	greater than.
$>=$	$\longrightarrow$	greater than or equal.
$<$	$\longrightarrow$	less than.
$<=$	$\longrightarrow$	less than or equal.

☒ `bool x;`

`x = 3 > 2;`

`cout << x;`

output $\Rightarrow$ 1.
value $\Rightarrow$ 1, true.

☒ `8 < 15;` $\longrightarrow$ true

`6 != 6;` $\longrightarrow$ false.

`'R' > 'T';` $\longrightarrow$ false.

* And, OR, Not operator :-

$\&\&$ $\longrightarrow$ And.
 $||$ $\longrightarrow$ Or.
 $!$ $\longrightarrow$ Not.

Exp 1

T
F
T
F

Exp 2

T
T
F
F

Exp 1 & 2

T
F
F
F

Exp 1 || 2

T
T
T
F

Exp 1

T
F

! Exp 1

F
T

* Example

1) $!('A' > 'B')$ $\longrightarrow$ True.

2) $(14 \leq 5) \&\& ('A' < 'B')$ $\longrightarrow$ False.

3) $(24 \geq 35) || ('A' > 'B')$ $\longrightarrow$ False.

4) $X = \text{true};$
 $Y = \text{false};$
 $Z = \text{false};$

$!(Y // Z) // X \rightarrow \text{true.}$

$Z \&\& Y \&\& X \rightarrow \text{false.}$

⑧ $5 // \underbrace{!X}_F \&\& \underbrace{Z}_F \rightarrow \text{true.}$

* Precedence :-

1) $!, +, -, \text{static_cast.}$
 $\rightarrow$ unary operator.

2) $*, /, \%$

3) $+, -$

4) $>, >=, <, <=$

5) $==, !=$

6) $\&\&$

7) $//$

* If statement :- key word.

1) if (grade >= 50)
 cout << "pass";] one statement.

2) if (grade >= 50);
 cout << "pass";] two separate statements.

3) if { grade >= 50 }
 cout << "pass";] error.

* If-else :-

~~if~~ if (Exp.)
 st 1;
 else
 st 2;

1) if (grade >= 50)
 cout << "pass";
 else
 cout << "fail";

2) if (grade >= 50)
 cout << "pass";
 else
 cout << "fail";
 cout << "You must take this course again";

يطاع هذه الجملة في حالة
الجاح وعبارة
وهذا خطأ.

Subject:

3) if (grade $\geq$ 50)

{

cout << "pass";

cout << "you can take the next course";

}

else

{

cout << "fail";

cout << "you must take this course again";

4) if (X > 0)

cout << "positive";

else

if (X < 0)

cout << "negative";

else

cout << "zero";

C++

~ ~ ~

G++

Subject:

* Question

```
X = 10;  
if (X == 5)  
    ++X;  
else  
    --X;  
cout << X;  
X = 6
```

```
X = 10;  
if (X == 5)  
    ++X;  
else  
    --X;  
cout << X;  
X = 9
```

⊗ cout << "In Gee, your \t" << age << endl;

طفاً

Gee, your 25

ممكن

syntax error. ← * إذا لم تكن age صيغة

C++

* Example) Write down a programme to read three integer numbers and print the maximum number of them.

```
#include <iostream.h>
int main()
{
    int x, y, z, Max;
    cout << "enter the first number ";
    cin >> x;
    cout << "enter the second number ";
    cin >> y;
    cout << "enter the third number ";
    cin >> z;
    if (x > y)
        Max = x;
    else
        Max = y;
    if (z > Max)
        Max = z;
    cout << "Max = " << Max << endl;
    return 0;
}
```

cout << "enter three numbers ";
cin >> x >> y >> z;

Subject: / /

* Switch Structure :-

If (X == 1)

cout << "one" ;

else if (X == 2) .

cout << "two" ;

else if (X == 3)

cout << "three" ;

else cout << "other numbers" ; .

* switch expression :- (using 4 key words) .

{

case value 1 : st1 ; st2 ; break ;

case value 2 : st3 ; st4 ; break ;

case value n : stn ; break ;

default : st ;

}

optional

(اختياري) يمكنه ان لا يوجد .
ويمكن ان لا يوجد .

يتم تنفيذ البرنامج حتى
يجد كلمة break

Subject:



(Ex)

switch (x)

{

case 1: cout << "one"; break;

case 2: cout << "two"; break;

case 3: cout << "three"; break;

default: cout << "Other numbers";

قسط
value
(x+1)
جواب

(Ex)

char ch;

cin >> ch;

switch (ch)

{

case 'R':

case 'r': cout << "Red"; break;

case 'G':

case 'g': cout << "Green"; break;

case 'O':

case 'o': cout << "Orange"; break;

}

* Exercises

①

alpha = 5;

beta = 2

cin >> beta;

112

switch (beta)

{

case 1: alpha + = 1;

Subject:

```
case 2 : alpha + = 2 ;  
case 3 : alpha + = 3 ;  
default ; alpha + = 4 ;  
}
```

$\alpha = 14$

② age = 50
switch (age > 21)
{
 True
 case true : cout << "legal Age"; break;
 case false : cout << "Below legal age"; break;
 default : cout << "Bad input";
}

Legal Age

③ int base = 2, result = 0;
switch (base)
{
 2
 case 1 : result = base; break;
 case 2 : result = base * base; break;
 case 3 : result = base * base * base; break;
 default : cout << "your base number is too large";
}

result = $2 * 2$

Subject:

④ $\text{int } a=3, b=4, c=5;$

$\text{if } (a > 2 \wedge c - b < 2) \text{ cout} \ll 'A';$

F $\text{if } (a/2 \stackrel{F}{=} 1 \parallel b \stackrel{F}{>} 6) \text{ cout} \ll 'B';$

$\text{if } (\neg(C == 5) \parallel a < b) \text{ cout} \ll 'C';$

F $\text{if } (\neg(a+3 > b-2 \wedge a == c-2)) \text{ cout} \ll 'D';$

AC

⑤ What is the final value of x if $x=9$.

$\text{if } (x \geq 0)$

$x = x + 1;$

$\text{if } (x \geq 1)$

$x = x + 2;$

$x = 12$

⑥ What is the value of "fee" if $\text{speed} = 95$.

$\text{if } (\text{speed} > 35)$

$\text{fee} = 20.00;$

$\text{else if } (\text{speed} > 50)$

$\text{fee} = 40.00;$

$\text{else if } (\text{speed} > 75)$

$\text{fee} = 60.00;$

$\text{fee} = 20.00$

⑦ num = 20

num 10 if (num <= 10)

if (num >= 0)

cont >> "num is between 0 & 10";
else cont >> "num is ~~between~~ greater than 10";

لا يطبع شيئاً

else ————— تابع أقرب إذا

⑧ if x = 9

if (x >= 0)

if (x >= 1)

x = x + 2;

else x = x + 3;

x = 11

if (x >= 0)

if (x <= 1)

x = x + 2

else x = x + 3;

x = 12

++

Subject: / /

* Chapter Four :-

* Repetition Structures :-

□ Counter Control :-

1) While 2) do/while 3) For

key words

△ While structure

initialization counter

while (exp)

{

st 1

st 2

st 3

⋮

increment / decrement

}

(ex) int i = 1 ;

while (i <= 10)

{ cout << i << " " ;

i++

}

2

Do-while structure :-

initialization counter

do

{ cout << i << " " ;

i++

}

while (i <= 10) ;

تطبع فراغ

* Do-while $\Rightarrow$ Minimum execution $\rightarrow$ 1 time.* while $\Rightarrow$ Minimum execution $\rightarrow$ 0 time.

3

For Structure :-

for (int i=1; i <= 10; i++)

{

cout << i << " " ;

}

④

①

②

③

* for loop is used to iterate a block of code a specific number of times.

* Chapter Four :-

* Repetition Structures :-

□ Counter Control :-

1) While 2) do/while 3) for

key words

△ While structure

initialization counter

while (exp)

{
st 1
st 2
st 3
⋮

increment / decrement

}

(ex) int i = 1;

while (i <= 10)

{ cout << i << " " ;
i++
}

②

Do-while structure :-

initialization counter

do

{ cout << i << " " ;

i++

}

while (i <= 10) ;

تطبیق خروج

* Do-while $\Rightarrow$ Minimum execution $\rightarrow$ 1 time.* while $\Rightarrow$ Minimum execution $\rightarrow$ 0 time.

③

For Structure :-

for (int i=1; i <= 10; i++)

{

cout << i << " " ;

}

④

①

②

③

* for loop is used when the number of iterations is known in advance.

Subject: _____

2] Sentienal Control :-

1) While

2) do, while.

sum = 0 , num = positive int.

* cin >> num;

while (num != -1) → *sentienal Control.*

```
{  
    sum += num;  
    cin >> num;  
}
```

6/4/3/-1

<u>num</u>	<u>sum</u>
6	6
4	10
3	13
-1	

* cin >> num;

while (num != -1)

```
{  
    cin >> num;  
    sum += num;  
}
```

6 4 3 -1

<u>num</u>	<u>sum</u>
6	
4	4
3	7
-1	6

* Exercises

1. $K = 1; j = 4;$
 for ($i = 1; i \leq 3; i++$)
 $K = K * i * --j;$
 cout << K;

<u>K</u>	<u>J</u>
1	4
3	3
12	2
36	1

??

2. $x = 3; y = 2; z = 5; S = 0;$

for (int $i = x; i \leq z; i++$)

$S = S + i + y;$

$y = y + 1;$

cout << S;

<u>i</u>	<u>x</u>	<u>y</u>	<u>z</u>	<u>S</u>
3	3	2	5	0
4		3		5
5				11
				18

3. $i = 5;$

for ($x = i; x \geq 2; x += -1$)

cout << x;

<u>x</u>
5
4
3
2

Subject: _____

```
[4] x = total = 5;
    while (x <= 10)
    {
        total += x;
        ++x;
    }
```

total
10
16
23
31
40
50

```
[5] int w = 5;
    while (5 (--w >= 0))
        cout << --w;
```

w
3/1/-1

```
[6] int w = 5;
    while (w-- >= 0)
        cout << --w;
```

w
4/3/2/1/0/-1

```
[7] int w = 5;
    while (--w >= 0)
        cout << w--;
```

w
4/3/2/1/0

```
[8] int w = 5;
    while (w-- >= 0)
        cout << w--;
```

w
5/5/5/... 5

Subject:

/ /

* Break & Continue :- Reserved words.

① * for ($i=1$; $i \leq 10$; $i++$)

```
{  
  if ( $i==5$ )  
    break;  
  cout << i;  
}
```

* Result)
1 2 3 4

② * for ($i=1$; $i \leq 10$; $i++$)

```
{  
  if ( $i==5$ )  
    continue;  
  cout << i;  
}
```

* Result)
1 2 3 4 6 7 8 9 10

③ * for ($i=1$; $i \leq 20$; $i++$)

```
{  
  if ( $i \% 5 == 0$ )  
    continue;  
  cout << i;  
}
```

* Result)
1 2 3 4 6 7 8 9 11 12
13 14 16 17 18 19

??

④

```
i = 1;  
while ( $i \leq 5$ )  
{  
  if ( $i \leq 3$ )  
    break; cout << i;  
}
```

* Result)

1 2 3

Subject:

??
⑤ $i=1;$
while ($i \leq 5$)
{ if ($i \leq 3$)
 ~~continue;~~
 cout << i; }
}

123

??
⑥ for ($i=1; i \leq 10; i++$)
{ if ($i==3 \parallel i==6 \parallel i==9$)
 continue;
 cout << i; }
}

369

* Nested loop :-

① for (int $i=1; i \leq 3; i++$)
{ for (int $j=1; j \leq 3; j++$)
 cout << 'x';
 cout << endl;
}

$j \leq i$
*
**

i	j	output
1	1	***
2	2	***
3	3	***
4	4	
	1	
	2	
	3	
	4	
	1	
	2	
	3	
	4	

Subject:

/ /

② $x = 1;$
 $\text{for}(i = 1; i \leq 2; i++)$
 $\text{for}(j = 0; j \leq 2; j++)$
 $x = x + j;$
 $\text{cout} \ll x;$

<u>i</u>	<u>j</u>	<u>x</u>
1	0	1
2	1	1
3	2	2
	3	4
	0	4
	1	5
	2	7
	3	

③ $i = 1;$
 $j = 2;$
 $\text{while}(i \leq 2)$
{ $\text{while}((j \geq 2) \&\&(j < 5))$
{ $\text{cout} \ll j;$
 $j++;$
}
 $i++;$
}
 $\text{cout} \ll "\backslash t" \ll i;$

<u>i</u>	<u>j</u>	<u>output</u>
1	2	234234...3
2	3	
3	4	
	5	

C++


```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
int n, i; sumE = 0; sumO = 0; num;
```

```
cout << "enter the number of integer";
```

```
cin >> n;
```

```
for (i = 1; i <= n; i++)
```

```
{ cin >> num
```

```
if (num % 2 == 0)
```

```
sumE += num;
```

```
else
```

```
sumO += num;
```

```
}
```

```
cout << "sum of even numbers is" << sumE;
```

```
cout << "sum of odd numbers is" << sumO;
```

```
return 0;
```

```
}
```

2 Write a c++ program to find the factorial of any number.

```
#include <iostream.h>
```

```
int main()
```

Subject:

```
{ int num;  
  int i;  
  int fact = 1;  
  cout << "Enter a number to find the factorial"; }
```

```
  cin >> num;  
  if (num < 0)  
    cout << "Error a negative number";  
  else  
    if (num == 0)  
      fact = 1;  
    else  
      for (i = num; i > 1; i--)  
        for (int i = 1; i <= num; i++)  
          fact * = i;  
  cout << "Fact of " << num << " is " << fact;  
  return 0;  
}
```

~ ~ ~
C++

Subject:

* Quizzes answers :-

(A) `#include <iostream.h>`
`int main()`
`{`
`int i, n, num, Even = 0, Odd = 0, zeros = 0;`
`cout << "Enter the number of integers";`
`cin >> n;`
`for (i = 1, i <= n, i++)`
`{`
`cin >> num;`
`if (num == 0)`
`zeros++;`
`if (num % 2 == 0)`
`Even++;`
`else`
`odd++;`
`}`
`cout << odd << even << zeros;`
`return 0;`
`}`

(B) `#include <iostream.h>`
`int main()`
`{`
`int grade, above 90 = 0, between 80-90 = 0, less 80 = 0;`
`cin >> grade;`
`while (grade >= 0)`
`{`
`if (grade >= 90)`
`above 90++;`
`}`

Subject:

else

if (grade $\geq$ 80)
 between 80-90 ++;

else

less 80 ++;

cin $\gg$ grade; $\longrightarrow$ serial control.

}

cout << "above 90 =" << above 90 << "between 80-90 =" <<
between 80-90 << "less 80 =" << less 80 << endl;
return 0;

}

③

A, B, C, D, F

کے مرتبہ

#include <iostream.h>

int main()

{ char grade;

int A=0, B=0, C=0, D=0, F=0;

cout << "Enter the grade ; '#' to end" ;

cin $\gg$ grade;

while (grade != '#')

{ switch (grade)

{ case 'A':

case 'a': A++; break;

case 'B':

case 'b': B++; break;

case 'C':

case 'c': C++; break;

Subject:

```
case 'D':  
case 'd': D++; break;  
case 'F':  
case 'f': F++; break;  
case ' ': case '\t': case '\n': break;  
default: cout << "illegal grade";  
}  
cin >> grade;  
}  
cout << A << B << C << D << F;  
return 0;  
}
```

*Ex) Write a C++ program to read a sequence of numbers until the sum of them exceeds 80. then evaluate and print the average.

```
#include <iostream>  
int main()  
{ int num, counter = 0, sum = 0;  
  float average;  
  while (sum <= 80)  
  { cin >> num;  
    sum += num;  
    counter++;  
  }  
  Ave = static_cast <float>(sum) / counter;  
  cout << Ave; return 0; }
```

Functions

user defined functions

Predefined Functions

math

string

#include <math>

<math.h>

* Math functions :-

1) $\text{ceil}(x) = \lceil x \rceil \Rightarrow \text{ceil}(9.2) = 10.0$

$\lceil x \rceil$ is the smallest integer greater than or equal to x

$\text{ceil}(-9.8) = -9.0$

$\text{ceil}(-8.2) = -8.0$

2) $\text{floor}(x) = \lfloor x \rfloor \Rightarrow \text{floor}(9.2) = 9$

$\text{floor}(-9.8) = -10.0$

$\text{floor}(-8.2) = -9.0$

3) $\text{fabs}(x) = |x|$

$\Rightarrow \text{fabs}(1) = 1$

$\text{fabs}(-1) = 1$

4) $\text{fmod}(x, y) = x \% y \Rightarrow \text{fmod}(5, 2) = 5 \% 2 = 1$

$$6) \text{ pow}(x, y) = x^y \Rightarrow \text{pow}(2, 3) = 2^3 = 8.$$

$$7) \text{ exp}(x) = e^x \Rightarrow \text{exp}(1) = e^1 = e = 2.718.$$

$$8) \log(x) = \ln x \Rightarrow \log(2.718) = \ln e = 1$$

$$9) \log_{10}(x) = \log x \Rightarrow \log_{10}(100) = 2.$$

$$10) \sin x, \cos x, \tan x$$

x (ثوابت، صيغرات، متغیبي)

*Example) $\text{int } x, y, z;$
 $x = 2; y = 3; z = 5;$

$$① \text{ sqrt}(z * (x + y))$$

$$= \text{sqrt}(5 * (2 + 3)) = \text{sqrt}(5 * 5) = 5$$

$$② \text{ sqrt}(\text{pow}(2, 3) + 8)$$

$$= \text{sqrt}(2^3 + 8) = \text{sqrt}(8 + 8) = \text{sqrt}(16) = 4.$$

C++

* Example) Max of 3 integer numbers.

include <iostream.h>

int maximum { int, int, int } ;

int main() parameters

{ int x, y, z, max;

cout << "Enter three integer numbers" ;

cin >> x >> y >> z ;

max = maximum (x, y, z) ;

cout << "maximum number is" << max ;

return 0 ;

}

leader function. int maximum (int a, int b, int c)

{ int m ;

if (a > b) .

m = a ;

else

m = b ;

if (c > m)

m = c ;

return m ;

}

body

Example) Max, Min for 3 integer numbers.

```
#include <iostream.h>
int maximum (int, int, int);
int minimum (int, int, int);
int main()
{ int x, y, z;
  cout << "Enter three integer numbers" << endl;
  cin >> x >> y >> z;
  cout << "minimum number is " << minimum(x, y, z) << endl;
  cout << "maximum number is " << maximum(x, y, z) << endl;
  return 0;
}
```

```
int maximum (int a, int b, int c)
{ int m;
  if (a > b)
    m = a;
  else m = b;
  if (c > m)
    m = c; return m; }
```

```
{ int m;
  if (u < v)
    m = u;
  else m = v;
  if (w < m)
    m = w; return m; }
```

Subject:

*Example) Write a c++ program that print the square of numbers from 1-10.

```
#include <iostream.h>
int square(int);
int main()
{
    for(int i=1; i<=10; i++)
        cout << square(i) << " ";
    return 0;
}

int square(int a)
{
    return a*a;
}
```

*Example) Write a function integer power (base, exp) that return value of base exponent.

```
#include <iostream.h>
int integerpower(int, int);
int main()
{
    int b, exp;
    cout << "Enter the base and the exponent respectively ";
    cin << b << exp;
    cout << b << "the base" to the power of " << exp << " is " << integerpower(b, exp);
    return 0;
}
```


Subject:

```
int integerPower (int base, int e)
{
    int product = 1;
    for (int i = 1; i <= e; i++)
        product * = base;
    return product;
}
```

H.W ① Write a complete C++ program that use a function sphere volume to evaluate the volume of a sphere according to its radius. $V = \frac{4}{3} \pi r^3$.

② Write a complete C++ program that use a function. Read a sequence of real numbers then round them using a function called "Round".
 $y = \text{Floor}(x + 0.5)$

③ Write a complete C++ program that use a function multiple that determine for a pair of integers whether a second integer is a multiple of the first. (a, b)

④ Find the prime numbers from 1-100 using function called "prime". by writing a complete C++ program.

Subject:

Ex

```
① int i=2;
   int j=3;
   while (i >= 1) {
       while (j > 1) {
           cout << (i+j)/2;
           --j;
       }
       --i;
       cout << (i+j)/2;
   }
```

220

```
② For (int i=4; i >= 1; i--)
{ for (K=1; K < i; K++)
    cout << " ";
  for (j=i; j <= 4; j++)
    cout << "*";
  cout << endl;
}
```

i	j	K	output
4	4	1	---*
3	5	2	--**
3	3	3	-***
4	1	4	****
		2	
		3	

$x=5, y=6, z=2$

③ $\text{floor} \left[(f \bmod (y \cdot x) + (\exp(\frac{y}{x}) + 1.5)) \right]$

$\text{floor}(1+2.5) = \text{floor}(3.5) = ③$

rand()

#include <cstdlib>

int i;

i = rand();

cout << i;

(K - n)

i = K + rand() % n - K + 1

① 0-1

i = rand() % 2

② 0-6

i = rand() % 7

③ 1-7

i = 1 + rand() % 7

④ 3-7

i = 3 + rand() % 5

*Example) Write a c++ program to print 5 random numbers from 1-10.

```
#include <iostream.h>
```

```
#include <cstdlib>
```

```
int main()
```

```
{ for( int i=1; i<=5; i++)
```

```
    cout << 1+ rand() % 10 << " ";
```

```
    return 0;
```

```
}
```

~~int seed;~~

~~cin >> seed;~~

→ int seed;

cin >> seed; srand (seed);

```
#include <iostream.h>
```

```
#include <cstdlib>
```

```
#include <ctime>
```

```
int main()
```

```
{
```

```
    srand (time(0));
```

```
    for (int i=1; i<=5; i++)
```

```
        cout << 1+ rand() % 10 << endl;
```

```
    return 0
```

```
}
```


* Chapter 6 : Part one Arrays :-

□ One dimensional arrays :-

* Declaration :

Data type ArrayName [index]

size (size { use })

* example) int x[5];



* access of array elements :

int x[5];

1) $x[2] = 3$

2) $i = 2;$

$x[i * 2 - 1] = 12$

$x[3] = 12$

0	1	2	3	4
0	0	3	12	0

3) $x[5] = 4$ (syntax error) "out of range"

4) $x[1] = x[3] - x[2];$

$= 12 - 3 = 9$

$\Rightarrow x[1] = 9$

5) $x[0] = x[1] = x[4] = 0$

Subject:

Ex ① $\text{int } y[6] = \{2, 4, 6, 8, 10, 12\}$

y	0	1	2	3	4	5
	2	4	6	8	10	12

② $\text{int } y[6] = \{6, 12\}$

y	0	1	2	3	4	5
	6	12	0	0	0	0

③ $\text{int } y[6] = \{0\}$

y	0	1	2	3	4	5
	0	0	0	0	0	0

④ $\text{int } z[] = \{1, 2, 3\}$

لم يحدد الحجم
فيكون الحجم حسب عدد
العناصر وضع (3)

0	1	2
1	2	3

⑤ $\text{const int size} = 10;$

$\text{int } S[\text{size}];$

$\text{for (int } i = 0; i < \text{size}; i++)$

$S[i] = 2 * i + 1;$

$S[0] = 1$

$S[1] = 3$

$S[2] = 5$

$S[3] = 7$

$S[4] = 9$

$S[5] = 11$

$S[6] = 13$

$S[7] = 15$

$S[8] = 17$

$S[9] = 19$

Subject:

5	0	1	2	3	4	5	6	7	8	9
1	3	5	7	9	11	13	15	17	19	

* int size;

cin >> size;

int S[size];

for (

يجب إدخال قيمة معرفة

⑥ Write a C++ program to declare a Float array numbers with size 20 then prompt the user to enter the values of the array then print it.

```
#include <iostream.h>
```

```
int main()
```

```
{ float numbers[20];
```

```
  for (int i=0; i<20; i++)
```

```
    cin >> number[i];
```

```
  for (int i=0; i<20; i++)
```

```
    cout << number[i] << " " << endl;
```

```
  return 0; }
```

إذا كانت البرنامج لايجاد مجموع مدخلات الصنوفة نرسله كالتالي:

```
float sum=0
```

```
for (int i=0; i<20; i++)
```

```
  sum += number[i];
```

```
cout << "sum of array numbers is" << sum;
```

⑦ Write a c++ program to find the sum of all odd & even numbers store in an integer array with size(100).

```
#include <iostream.h>
int main()
{
    int sumE = 0, sumO = 0;
    int x[100];
    for (int i = 0; i < 100; i++)
    {
        if (x[i] % 2 == 0)
            sumE += x[i];
        else
            sumO += x[i];
    }
    cout << "sum of even numbers is" << sumE;
    cout << "sum of odd numbers is" << sumO;
    return 0;
}
```

⑧ $\text{int } x[20] = \{2, 4, 6, 8, 10, 12\}$
 $\text{int } y[20];$

i) $y = x;$ (illegal)

ii) $\text{for } (i = 0; i < 20; i++)$ (legal)
 $y[i] = x[i];$

iii) $\text{cin} >> y;$ (illegal)

Subject:

iii) for (i=0; i<20; i++) (legal)
cin >> y[i];

* Passing array to functions :-

prototype :- int readarray (int[], int)

int readarray (int x[], int S)
readarray (x, S)

Ex Write a C++ program to do the following :-

- 1) Create function "read array" to read an int array of size 10.
- 2) Create function "print array" to print an int array of size 10.
- 3) Create function "sum" to find the summation of array elements.

Sol #include <iostream.h>

void readarray (int [], int);

void printarray (int [], int);

int sum (int [], int);

int main()

{ const int S=10;
int a[S];

Subject:

/ /

```
cout << "Enter 10 integer numbers";  
read array (a, s);
```

```
cout << "The sum of the array element is <<  
sum(a, s);
```

```
cout << "This is the element of the array";  
print array(a, s);  
return 0;
```

```
}
```

```
void read array (int x[], int size)
```

```
{  
    for (int i=0; i<s; i++)  
        cin >> x[i];
```

```
}
```

```
int sum(int x[], int size)
```

```
{  
    int sum=0;  
    for (int i=0; i<size; i++)  
        sum += x[i];  
    return sum;
```

```
}
```

```
void print array (int x[], int size)
```

```
{  
    for (int i=0; i<size; i++)  
        cout << x[i] << " ";
```

```
}
```

Note

Array called
by reference
not by value

Subject:

* Array of Characters : "STRING" :-

char Name[10]; ' \0 '

0	1	2	3	4	5	6	7	8	9
'H'	'E'	'L'	'L'	'O'	'\0'				

char Name[6]; ' \0 '

↓
(\0) ه موقع السادس ل

Name = "Hello"; (\0 by default)

Name = { 'H', 'e', 'l', 'l', 'o', '\0' }

char Name[] = "Wafaa";

↑ 5
6 ↓ \0

* To Read or Print string :-

i) cin >> Name; OR for (int i=0; i<size; i++)
cin >> Name[i];

ii) cout << Name; OR for (int i=0; i<size; i++)
cout << Name[i];

Subject:

Ex ~~2~~ char $S_1[16] = \text{"welcome to you"}$

1) char $S_2[16] = S_1;$

2) $S_2[0] = S_1[0];$

$S_2[1] = S_1[1];$

⋮

$S_2[16] = S_1[16];$

3) for (int $i=1; i < 16; i++$)

$S_2[i] = S_1[i];$

* Function s :-

<cstring>

1) strcpy(S_1, S_2);

2) strncpy(S_1, S_2, n);

3) strcmp(S_1, S_2);

- if $S_1 = S_2$ (0)
- if $S_1 < S_2$ (<0)
- if $S_1 > S_2$ (>0)

4) strncmp(S_1, S_2, n);

5) strcat(S_1, S_2);

6) strncat(S_1, S_2, n);

7) strlen(S_1);

ويعرج (int number)

عدد داخل string (char)

Subject:

Ex

① `int x;`

`S1 = "Hello";`

`S2 = "Hellw";`

`x = strcmp(S1, S2);` → `x < 0`

`x = strcmp(S1, S2, 4);` → `x = 0`

② `S1 = "welcome";`

`S2 = "to you";`

`cout << strcat(S1, S2);`

لا يوجد فراغ

* `welcometo you`

`cout << strcat(S1, S2, 2);`

↓
من S₁ و S₂ (أول 2 نقطه)

* `welcometo`

`cout << strlen(S1);` → **7**

③ `char studentName[21];`

`char My Name[16];`

`char Your Name[16];`

`int length;`

1) `strcpy(My Name, "Rahaf Shaheen");`
`My Name = "Rahaf Shaheen";`

Subject:

2) strcpy (My Name, "Rahaf Shaheen", 5) ;

My Name = "Rahaf" ;

3) length = strlen("sunny day") ;

length = 9 .

4) strcpy (Your Name, "Ahmed Haddad") ;

Your Name = "Ahmed Haddad" ;

5) strcpy (Student Name, Your Name, 5) ;

Student Name = "Ahmed" ;

6) strcmp ("saleem", "saleq") ; < 0

* الحرف الأكبر من (space)

7) strcmp ("saleem", "saleq", 4) ; = 0

~~~~~



## \* Two-Dimensional Arrays :-

### Declaration :

char array 1 [row][col]

char array 1 [3][4]

|   | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 |   |   |   |   |
| 1 |   |   |   |   |
| 2 |   |   |   |   |

\* ① char array 2 [2][3] = { {2, 4, 6}, {10, 20, 30} }

|   | 0  | 1  | 2  |
|---|----|----|----|
| 0 | 2  | 4  | 6  |
| 1 | 10 | 20 | 30 |

\* array 2 [0][0] = 2

array 2 [0][1] = 4

array 2 [0][2] = 6

\* ② char array 2 [2][3] = { 2, 4, 6, 10, 20, 30 } ;

|   | 0  | 1  | 2  |
|---|----|----|----|
| 0 | 2  | 4  | 6  |
| 1 | 10 | 20 | 30 |

تعتبر بالصفوف بهذه الطريقة تبدأ صفياً

① + ② تعتبر لنفسه بالصفوف

Subject: .....

③ char array 3 [2][3] = { {<sup>row</sup>2, 4}, {<sup>row</sup>10} }

|   | 0  | 1 | 2 |
|---|----|---|---|
| 0 | 2  | 4 | 0 |
| 1 | 10 | 0 | 0 |

\* Processing of Two-Dimensional Array :-

- 1) Entire array.
- 2) Row processing.
- 3) Col processing.

\* suppose the following declaration :-

```
const int rowsize = 3;  
const int colsize = 4;  
int Number [rowsize][colsize];  
int row;  
int col;
```

1) Initialization of row number three to zero.

0-2 0-3

```
Number[2][0] = 0;  
Number[2][1] = 0; OR for(col=0; col<colsize; col++)  
Number[2][col] = 0;  
Number[2][2] = 0;  
Number[2][3] = 0;
```



Subject: .....

2) Initialization of col number 2 to zero.

`col = 1;`

`for (row = 0; row < rowsize; row++)`

`Number[row][col] = 0;`

`OR`

`for (row = 0; row < rowsize; row++)`

`Number[row][1] = 0;`

3) Initialization of entire array to zero.

`for (row = 0; row < rowsize; row++)`

`for (col = 0; col < colsize; col++)`

`Number[row][col] = 0;`

Input array :-

`for (row = 0; row < rowsize; row++)`

`for (col = 0; col < colsize; col++)`

`cin >> number[row][col];`

Output array :-

`for (row = 0; row < rowsize; row++)`

`{ for (col = 0; col < colsize; col++)`

`cout << Number[row][col] << " ";`

`cout << endl; }`

مخرجات  
صف  
صف

Subject: .....

\*Ex

|   | 0  | 1  | 2  | 3  |
|---|----|----|----|----|
| 0 | 2  | 4  | 6  | 8  |
| 1 | 10 | 20 | 30 | 40 |
| 2 | 3  | 6  | 7  | 1  |

- ① Sum of row 1 is 20.  
Sum of row 2 is 100.  
Sum of row 3 is 17.

- ② Sum of col 1 is 15.  
Sum of col 2 is 30.  
Sum of col 3 is 43.  
Sum of col 4 is 49.

Sol ① sum of each row.

② ③  
for (row = 0; row < rowsize; row++)  
{ sum = 0;  
 ③  
 for (col = 0; col < colsize; col++)  
 sum += Number[row][col];  
 cout << "sum of row" << row+1 << " is " << sum; }

② sum of each column:



```

for(col=0; col < colsize; col++)
{
    sum=0;
    for (row=0; row < rowsize; row++)
        sum += Number[row][col];
    cout << "sum of col" << col+1 << "is" << sum; }

```

لو عرفنا المجموع في البداية  
لأصبح المجموع في كل مرة صفرًا  
هو مجموع آخر عمود فقط

(هذه هي فكرة طباعة مجموع آخر صف أو آخر عمود في مصفوفة)

\* Square array :-

int array 1[3][3]

|   | 0  | 1  | 2  |
|---|----|----|----|
| 0 | 00 |    |    |
| 1 |    | 11 |    |
| 2 |    |    | 22 |

if (row+col == row-1)

if (row == col)

row < col "upper triangle"  
row > col "lower triangle"



Subject: .....

/ /

Diagonal :  $\text{row} = \text{col}$   
opposite diagonal  $\text{row} + \text{col} = \text{row} - 1$

lower triangle with Diagonal  $\text{row} \geq \text{col}$   
upper  $\text{row} \leq \text{col}$

The End