Prep Smart. Score Better.

Programming in C Language

ABOUT ME : NAVNEET GUPTA

- 8 years teaching experience.
- AIR 92 in GATE 2008
- Qualified UGC-NET 2012, Raj.-SET 2012, CSIR-Recruitment-Exam in 2011
- Achieved 3rd Rank in NPTEL-DBMS Course
- Achieved Silver Medal in CSIR on ERP Project in 2013
- Area of Expertise : DBMS, Programming, Algorithms, Discrete Maths, Computer Networks, Operating system

gradeup
Prep Smart. Score Better.

Variables

A variable is used to store a piece of data for processing. It is called variable because you can change the value stored.

gradeup
Prep Smart. Score Better.

Expressions

gradeup
Prep Smart. Score Better.

Keywords in C

Data types in C Language

C language supports 2 different type of data types:

1.Primary data types:

These are fundamental data types in C namely integer(int), floating point(float), character(char) and void.

2.Derived data types:

Derived data types are nothing but primary datatypes but a little twisted or grouped together like **array**, **stucture**, **union** and **pointer**. These are discussed in details later.

gradeup
Prep Smart. Score Better.

Integer type

int → 2 Bytes (on 16-bit machine)

2 Bytes = 16 bits

2^{16} = 65536 / 2

−32,768 to +32767

} Range

gradeup
Prep Smart. Score Better.

Floating point type

*) It takes 4 Bytes

* Float value is stored in scientific notation

mantissa Exponent

gradeup
Prep Smart. Score Better.

Character type

(*) It takes only 1 Byte

1 Byte = 8 bit

$2^8 = 256$

-128 to +127

gradeup
Prep Smart. Score Better.

unsigned
only +ve values

signed
both -ve & +ve values

void type

*) It is used with pointers & return type of the function.

unsigned int a;
↓
0 to 65535

unsigned char m;
↓
0 to 255

gradeup
Prep Smart. Score Better.

6. (UGCNET-Dec2013-II-17) Which of the following has compilation error in C ?

(A) int n = 32;

(B) char ch = 65;

(C) float f= (float) 3.2;

(D) none of the above

type - casting

gradeup
Prep Smart. Score Better.

IOC

UGCNET-Dec2012-II-7)The ‘C’ language is

(A) Context free language (B) Context sensitive language

(C) Regular language (D) None of the above

gradeup
Prep Smart. Score Better.

(UGCNET-June2012-II-38)

printf("%c", 100);

(A) prints 100

(B) prints ASCII equivalent of 100

(C) prints garbage

(D) none of the above

gradeup
Prep Smart. Score Better.

Operators in C Language

! = Negation → T to F, F to T

~ = 1's complement

& = address of

Operator	Name	Operators	Associativity
①	postfix	[], (), →, ., ++, --	Left to Right
②	Unary operator	++, --, !, ~, (type), &, sizeof	Right to left
③	Arithmetic	*, /, %, +, -	Left to right

b = c + (a - d)

b = a + c * d

sizeof = no. of bytes

gradeup
Prep Smart. Score Better.

Operators in C Language

④ Shift Operator	>> , <<	Left to right
⑤ Comparison	>, >=, <, <= ✓ == , != ✓	,,
⑥ Bit-wise	& , | , ^	n

int a = 5;
a << = 2;
a = ?
shortcut
n * 2^2
⇒ 5 * 2 * 2 = 20

a = 5;
a >> = 2;
a = ?
a = 1

a = 500;
a >= 3;
a = ?
500 / 2^3 = ?

gradeup
Prep Smart. Score Better.

& , | , ^ ex-OR

int a = 5, b = 10, c;

c = a & b; c = 0

Operators in C Language

Logical	&&, ||	Left-to-right
Conditional	? : (ternary)	right to left
Assignment	+=, -=, =, >>=, <<=	right to left

c = (a >= b) ? a+b : a*b;

expression

T

F

C = 50 A

5 >= 10 ? 5+10 : 5*10

F

gradeup
Prep Smart. Score Better.

T = 1 (any non-zero value)
F = 0

2. main ( )

{

Int x = 10, y = 20, z = 5;

Int i;

i = x < y < z ;

Print f (“% d”, i);

}

x < y < z

10 < 20 < 5

1 < 5

i = 1

1

A. 1 B. 0 C. Error D. Not

gradeup
Prep Smart. Score Better.

```
3. main ( )
{
Int I = –3, J = 2, K = 0, m;
M = ++I && ++J || ++K;
}
```

i = ? J = ? K = ? m = ?

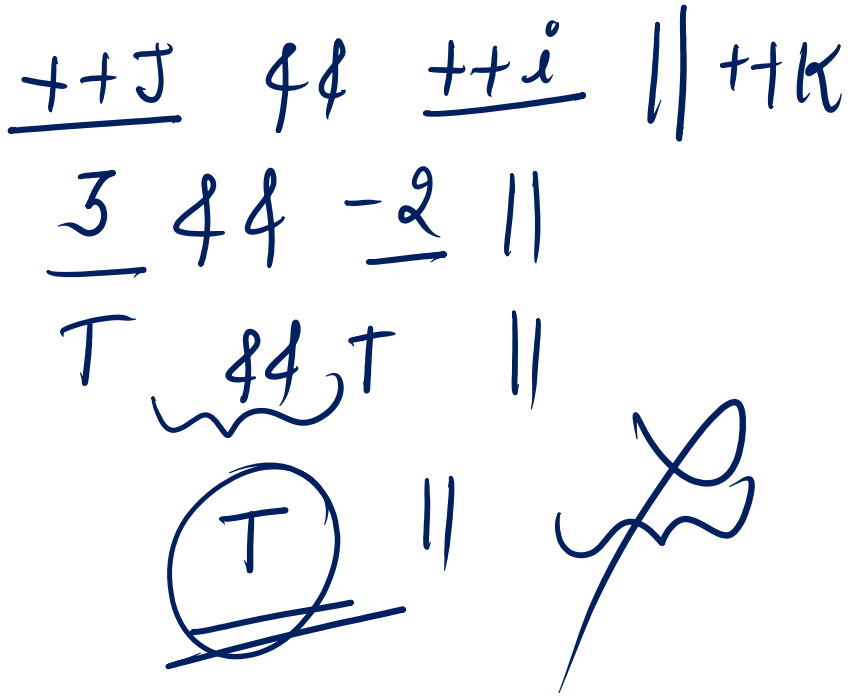
++i : prefix ⇒ first increment then assign

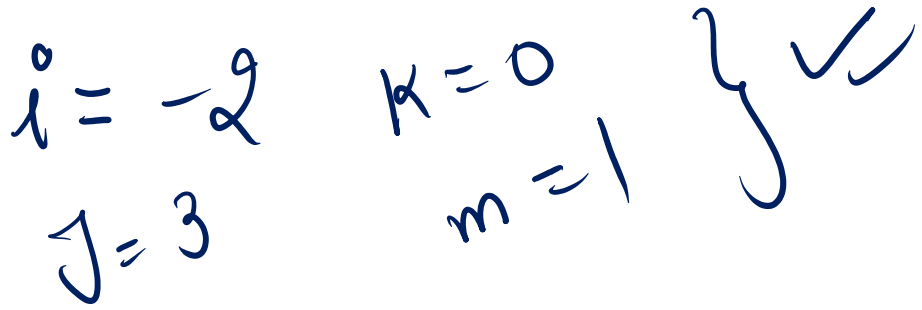
i++ : postfix ⇒ first assign then increment

gradeup
Prep Smart. Score Better.

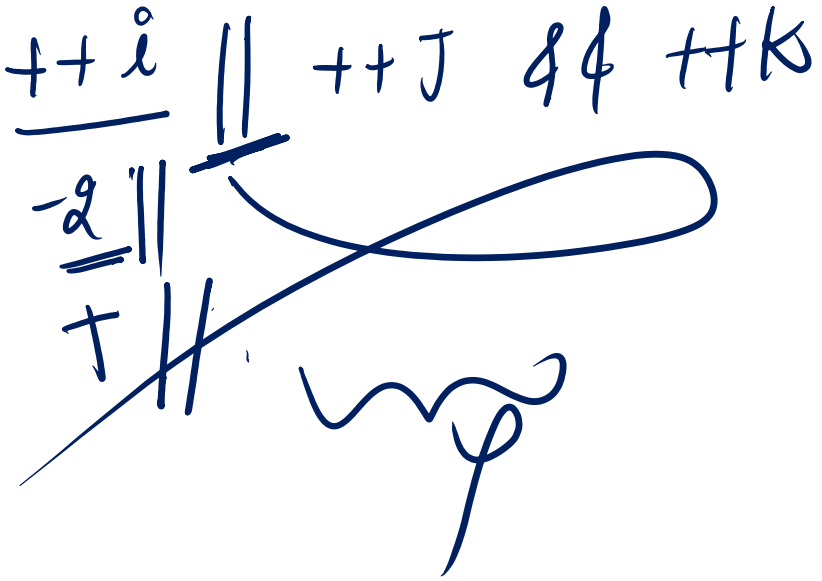
```
4. main ( )
{
Int I = –3, J = 2, K = 0, m;
m = ++J && ++i  || ++ k
}
```

i = ? J = ? K = ? m = ?





gradeup
Prep Smart. Score Better.



5. main ( )

{

Int I = –3, J = 2, K = 0, m;

m = ++I || ++J && ++K

}

i = ? J = ? K = ? m = ?

gradeup
Prep Smart. Score Better.

F=0
Revision

6. main ( )

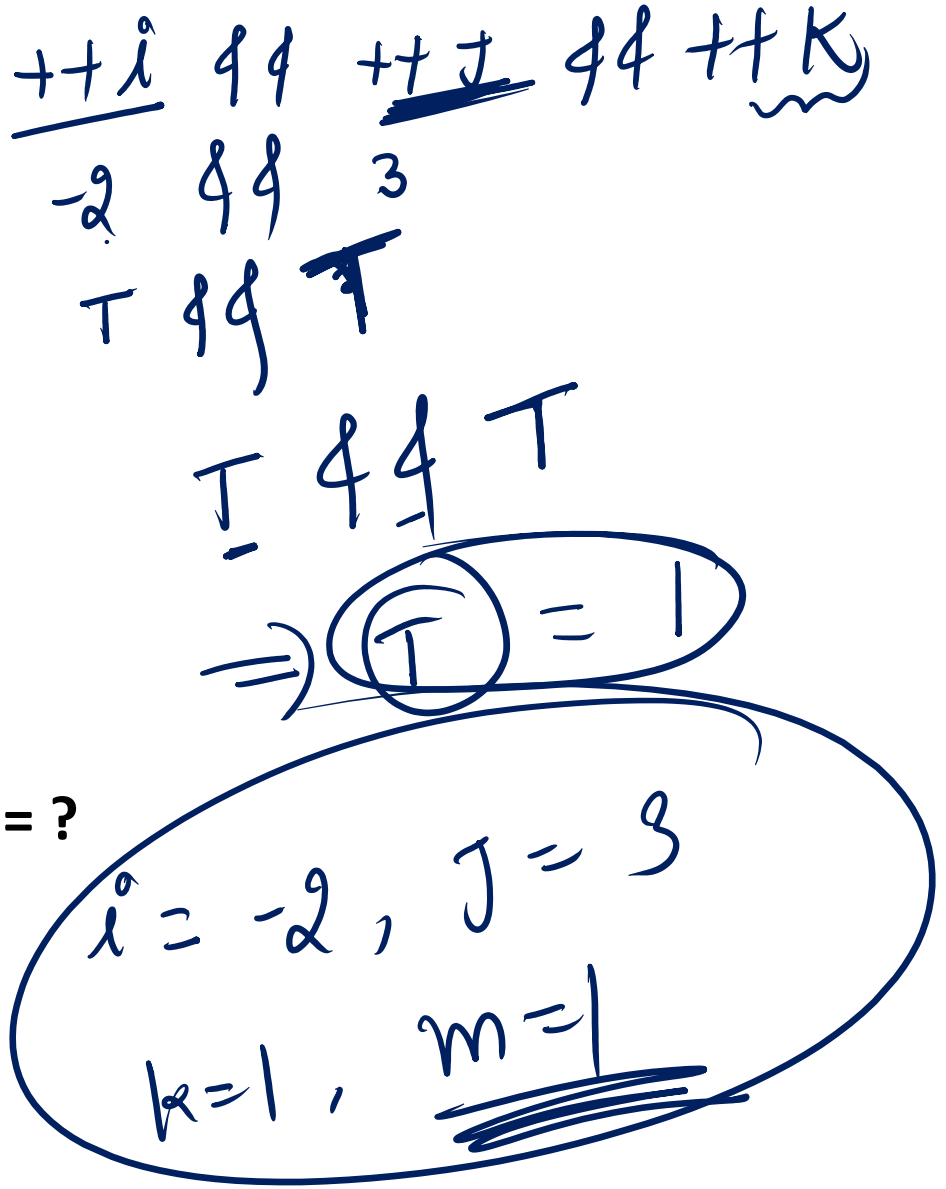
{

Int I = –3, J = 2, K = 0, m;

m = ++i && ++J && ++K

}

i = ? J = ? K = ? m = ?



gradeup
Prep Smart. Score Better.

(UGCNET-June2010-II-11) The statement

print f (" % d", 10 ? 0 ? 5 : 1 : 12);

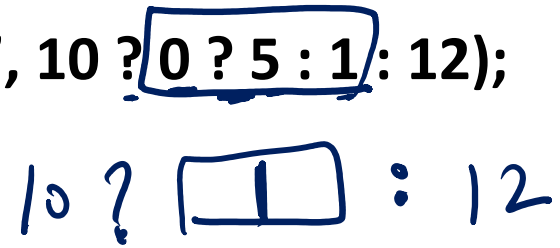
will print

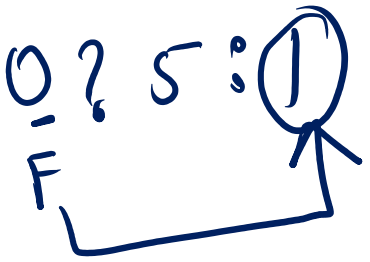
(A) 10

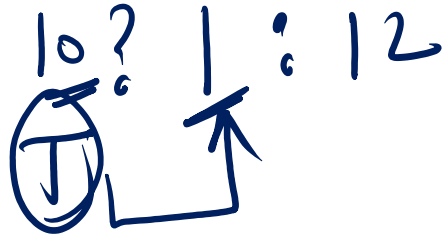
(B) 0

(C) 12

(D) 1







gradeup
Prep Smart. Score Better.

Basic 'C'

Decision making in C

- if statement
- switch statement
- conditional operator statement (? : operator)
- goto statement

gradeup
Prep Smart. Score Better.

Switch statement in C

```
switch(A)
{
case
    =
    =
    =
can :


}
```

gradeup
Prep Smart. Score Better.

```
1. main ( )
{
Int I = 4, J = 2;
Switch(i)
{
Case i :
Print f (“Hello”)
break;
Case j :
Print f (“Hi”);
Break;
}
}
```

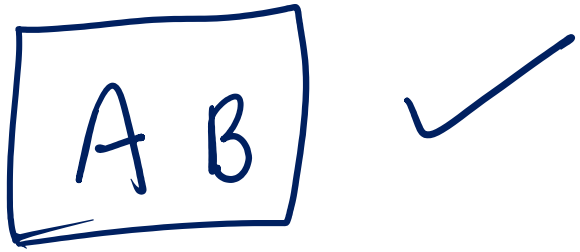
o/p Error

variable

only constant or expressions are allowed not variable

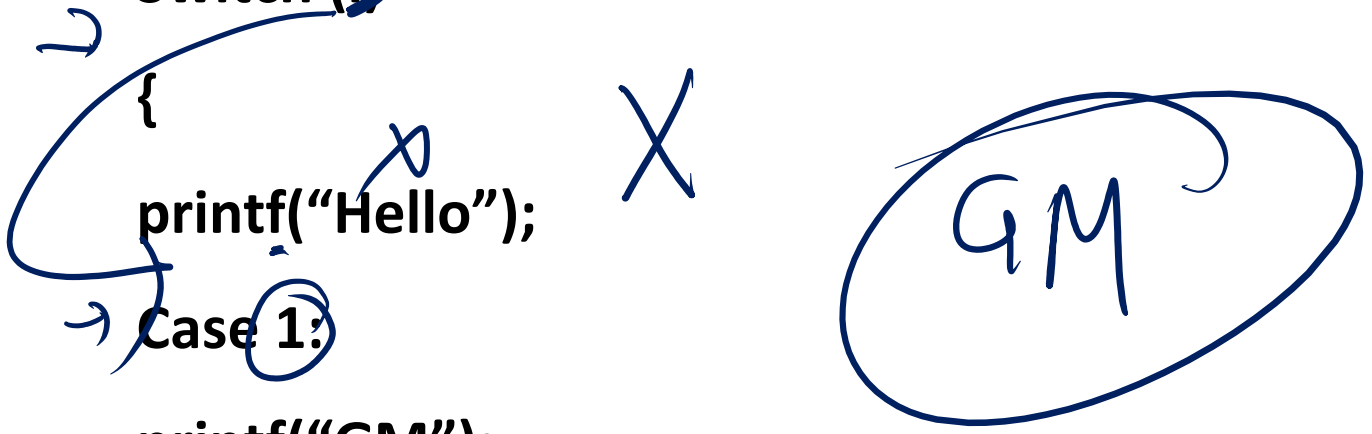
gradeup
Prep Smart. Score Better.

```
2. main ( )
{
Int i = 4;
Switch (i)
{
Default: print f(“A”);
Case1 : print f(“B”);
break;
Case2: print f(“C”)
Break’;
Case3: print f(“E”);
}
```



gradeup
Prep Smart. Score Better.

```
4. main ( )
{
Int i = 1;
Switch (i)
{
printf(“Hello”);
Case 1:
printf(“GM”);
        break;
Case 2:
printf(“GN”);   break; } }
```



gradeup
Prep Smart. Score Better.

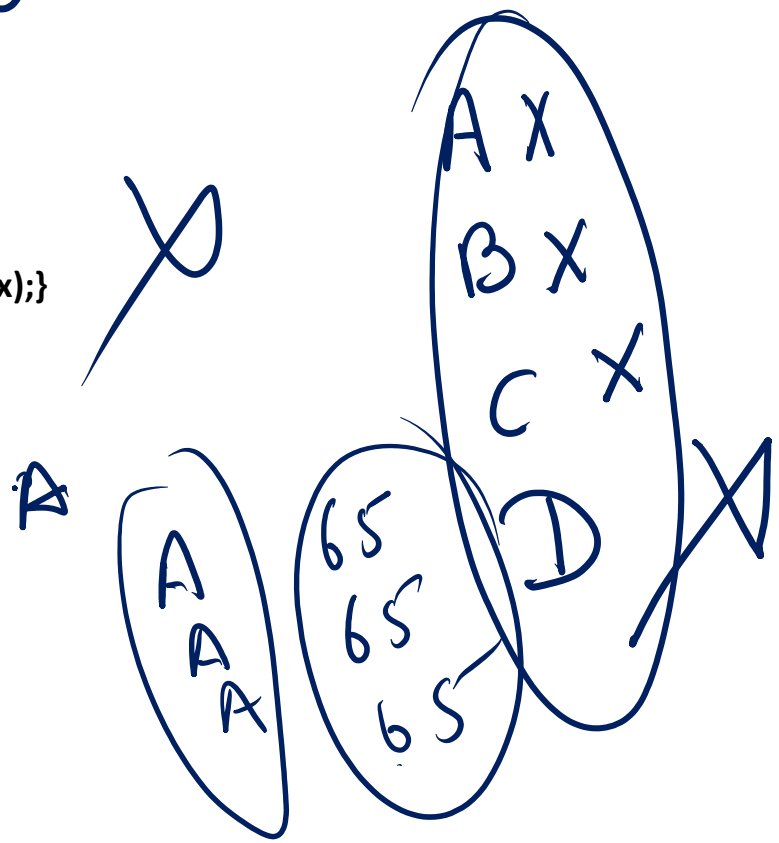
23. (UGC net Paper II December 2007 No 13)Which of the following is a valid C code to print character 'A' to 'C' ?

(A) x='A';
switch(x)
{case 'A'=printf ("%d\n", x);
....
case 'C'=printf ("%d\n", x);
}

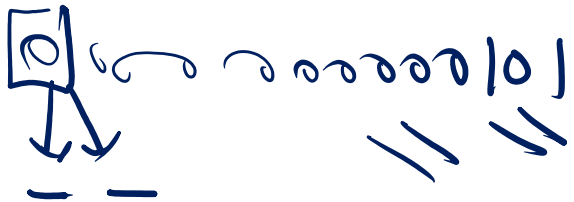
(B) x='A';
switch(x)
{case 'A'<=x <='C' : printf ("%d\n", x);}

(C) x='A';
switch(x)
{
case 'A' : printf ("%d\n", x);
break;
case 'B' : printf ("%d\n", x);
break;
case 'C' : printf ("%d\n", x);
break;
}

(D) x='A';
switch(x)
{
case 'A'=printf ("%d\n", x);
case 'B'=printf ("%d\n", x);
case 'C'=printf ("%d\n", x);
}



gradeup
Prep Smart. Score Better.



(UGCNET-june2007-12) In case of right shift bitwise operator in 'C' language, after shifting n bits, the left most n bits:

(A) are always filled with zeroes

(B) are always filled with ones

(C) are filled with zeroes or ones and is machine dependent

(D) none of the above

gradeup
Prep Smart. Score Better.

26. What is the output of the following ‘C’ program?

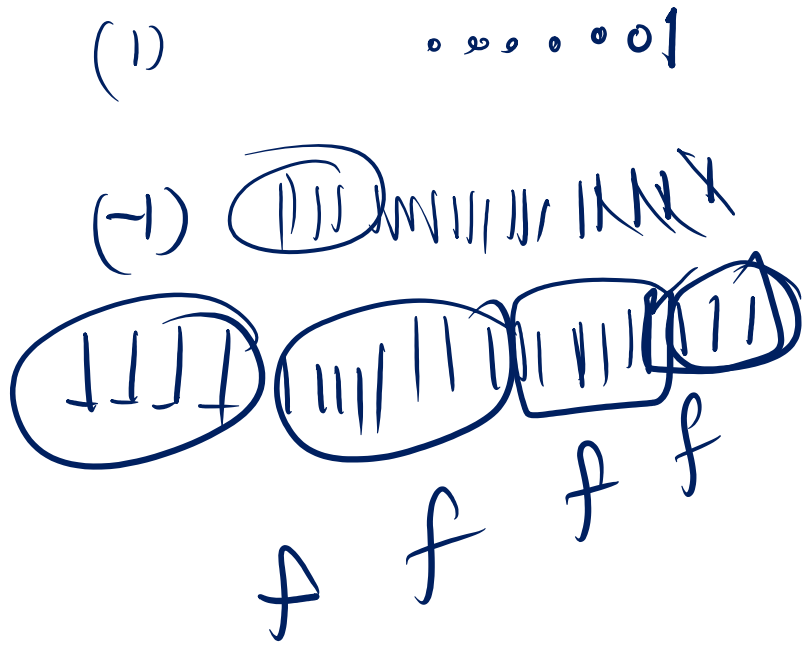
```
main()
{print f(“%x”,-1>>4);}
```

(A) ffff

(B) 0fff

(C) 0000

(D) fff0



gradeup
Prep Smart. Score Better.

3.14f → float
double

What is the output of the following C program main()

{

printf("%d%d%d", sizeof(3.14f), sizeof(3.14), sizeof(3.141));

}

(A) 4 4 4

(B) 4 8 8

(C) 8 4 8

(D) 8 8 8

gradeup
Prep Smart. Score Better.

Find the output of the following "C" code:

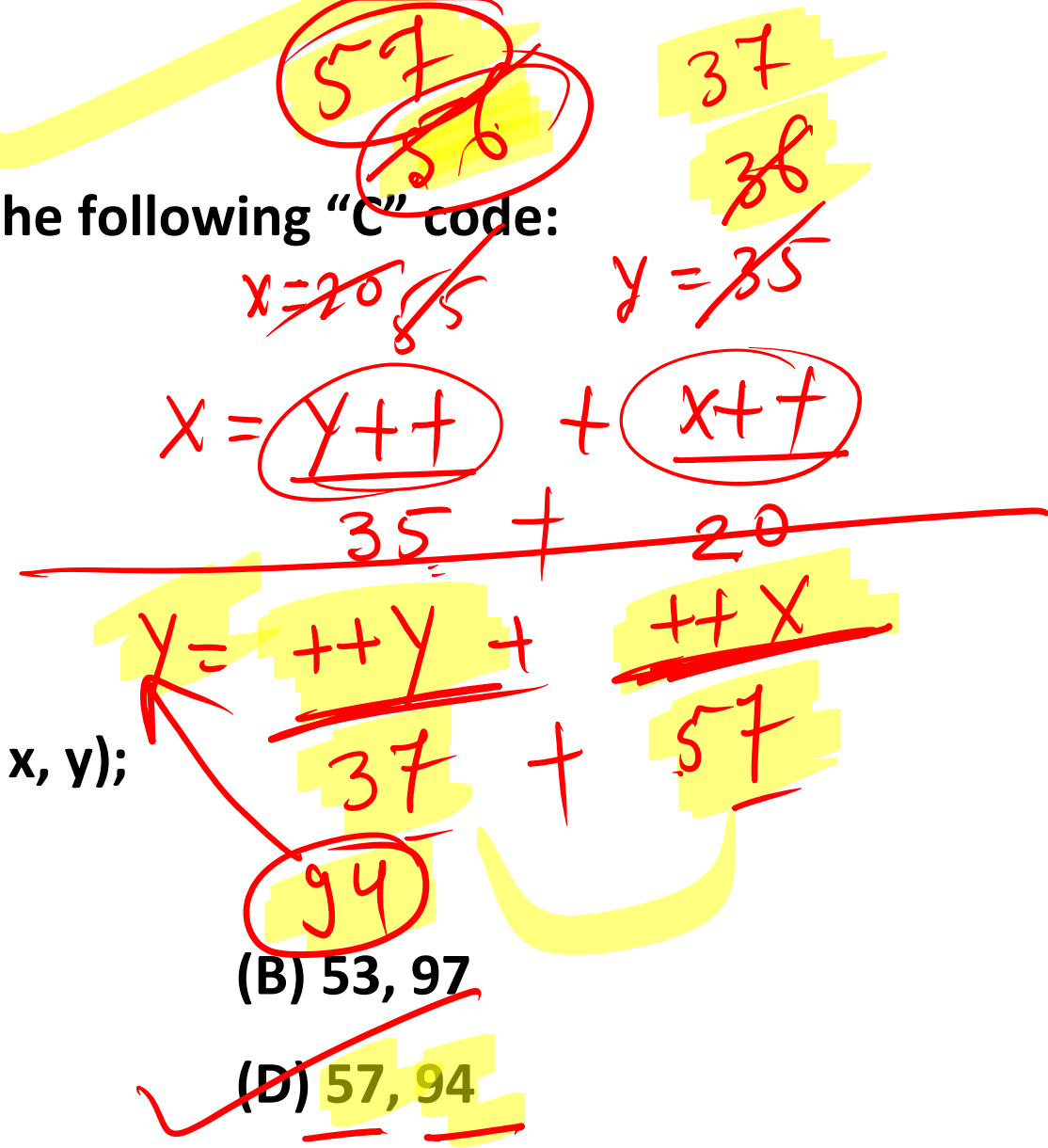
```
Main ( )
{ int x = 20, y = 35;
x = y++ + x++;
y = ++y + ++x;
printf ("%d, %d\n", x, y);
}
```

(A) 55, 93

(B) 53, 97

(C) 56, 95

(D) 57, 94



What are Loops?

A **Loop** executes the sequence of statements many times until the stated condition becomes false. A loop consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false. The purpose of the loop is to repeat the same code a number of times.

Types of Loops in C

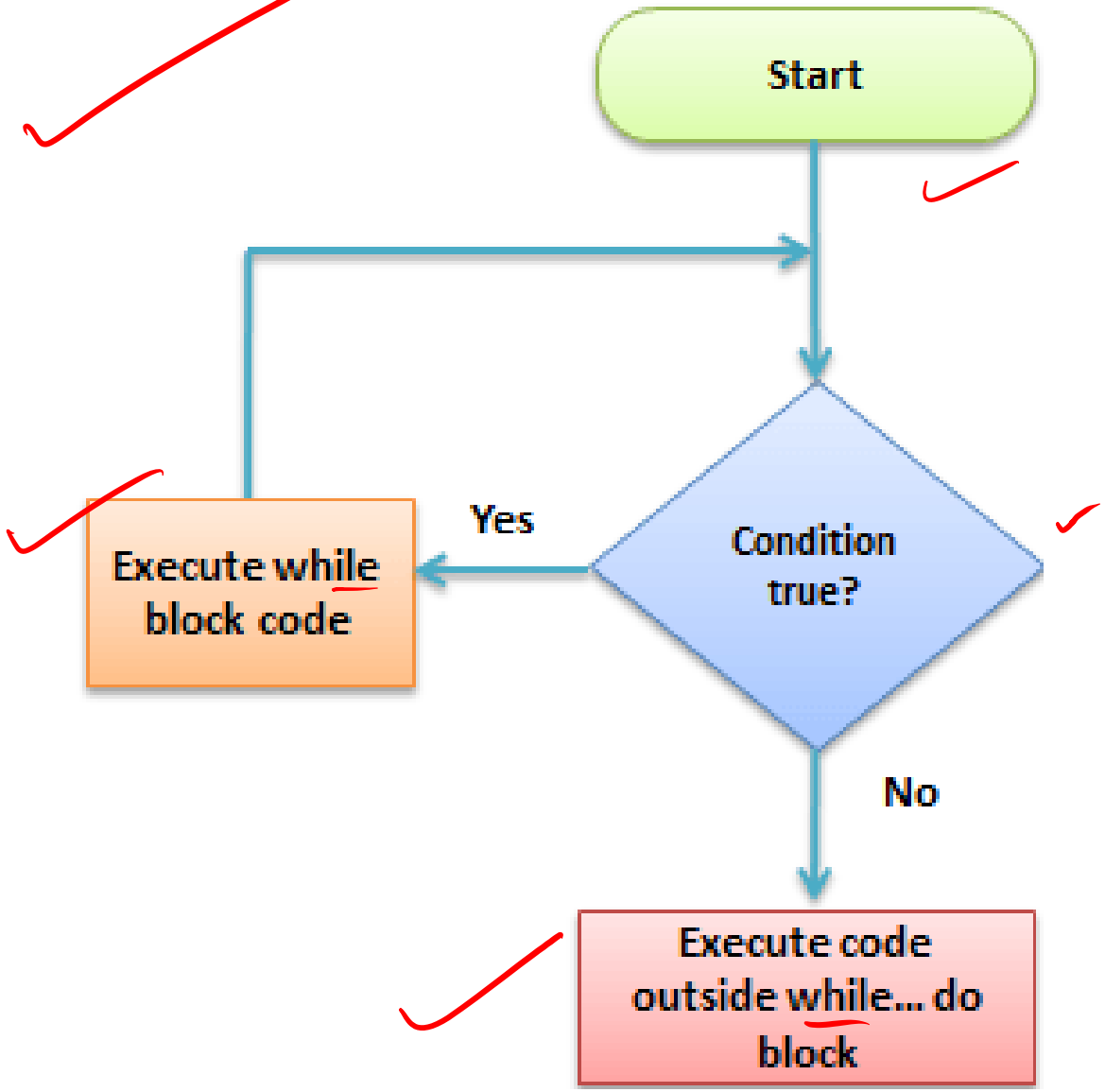
Depending upon the position of a control statement in a program, looping in C is classified into two types:

(i) Entry - Controlled Loop → for, while

Condition is put on entry point.

(ii) Exit - Controlled Loop → do-while

Condition is put on exit point.


Start
Condition true?
Yes
Execute while block code
No
Execute code outside while... do block

C' programming language provides us with three types of loop constructs:

(i) for loop

(ii) while loop

(iii) do - while loop

For loop

Syntax ①

```
for (initial value; condition; incrementation or decrementation )
{
        statements;
}
```

for (; ;)

infinite loop

gradeup
Prep Smart. Score Better.

8 PM