



Prep Smart. Score Better.



# gradeup

Prep Smart. Score Better.

**C Language- Functions**

# 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 **3<sup>rd</sup> 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



## Decision making in C

- if statement ✓
- ✓ • switch statement ✓
- conditional operator statement (? : operator) ✓
- goto statement ✗
- ✓ **loops** ✗

Case Sensitive

for

for

## Switch statement in C

syntax

switch (variable)

{

case 'A' :

≡

case 'B' :

≡

}

}

```
1. main ( )  
{  
Int i = 4, j = 2;  
Switch(i)  
{
```

Case i :

Print f ("Hello")  
break;

Case j :

Print f ("Hi");  
Break;

}

}

error

1 2 'A'  
1+2+3  
↑ expression  
Constant

2. main ( )

{

Int i = 4;

Switch (i)

{

Default: print f("A");

Case1 : print f("B");

break;

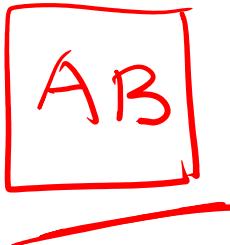
Case2: print f("C")

Break';

Case3: print f("E");

}

}



3. main ( )

{

Int i = 1;

Switch (i)



{

Case 1:

Print f("GM");

(break;

Case 1 \* 2 + 4 :

print f("GN");

Break;

}

## 4. main ( )

{

Int i = 1;

Switch (i)

{

Print f("Hello");

Case 1: print f("GM");

break;

Case 2: print f("GN");

break;

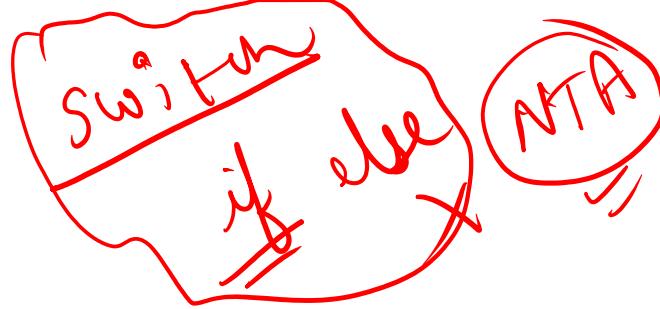
}

}

default : pf( - );

Case 1:





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

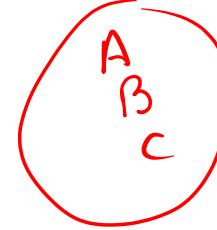
(A) ~~x='A';~~  
switch(x)  
{case 'A'=printf ("%d\n", x);  
....  
case 'C'=printf ("%d\n", x);  
}

(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);  
}

:



65

No ~~as~~  
option matches



(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 X
- (B) are always filled with ones X
- (C) are filled with zeroes or ones and is machine dependent
- (D) none of the above

26. What is the output of the following 'C' program?

main()

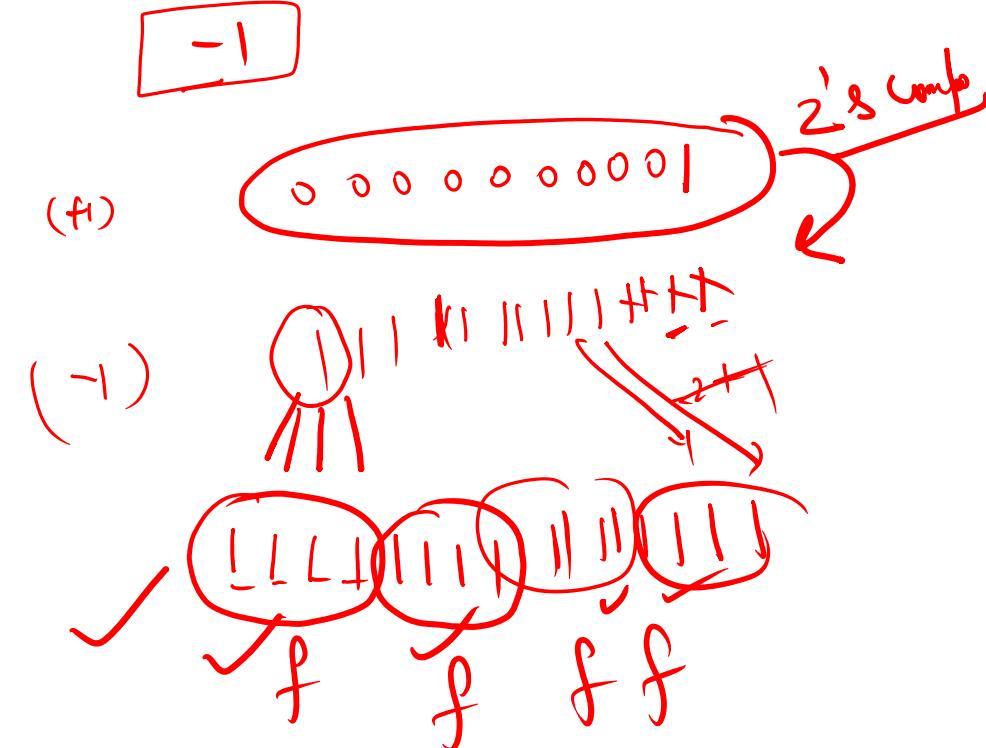
{printf("%x",-1>>4);}

(A) ffff

(B) 0fff

(C) 0000

(D) fff0



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

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

*no. of bytes*

*float*

*double*

(A) 4 4 4

~~(B) 4 8 8~~

~~(C) 8 4 8~~

~~(D) 8 8 8~~

- Post incr ✓
- 1) Value used / point assign
- 2) Value incremented
- Pre incr
- 1) value incre  
2) " assis'to use  
print
- 1) Value used / point assign
- 2) Value incremented

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

(C) 56, 95

$$\begin{aligned}
 x &= \cancel{y++} + \cancel{x++} \\
 &\quad 35 + 20 - 55 \\
 y &= \cancel{35} \cancel{36} 37 \\
 &= 37 + 57 \\
 x &= 57
 \end{aligned}$$

$$\begin{aligned}
 y &= \cancel{++y} + \cancel{++x} \\
 &\quad 37 + 57 \\
 &= 94
 \end{aligned}$$

(B) 53, 97

✓ (D) 57, 94

## Functions

user-defined function

- (i) function declaration
- (ii) function use / call
- (iii) function definition

Examples.1

Calling function

int add(int, int); ← fun decl.

→ main()

int a=10, b=20;  
int c;

Actual parameter

c = add(a, b);  
printf("%d", c);

function use  
formal parameters.

int add(int x, int y);  
return x+y;

x → 10  
y → 20

y

~~Examples 2~~Recursion

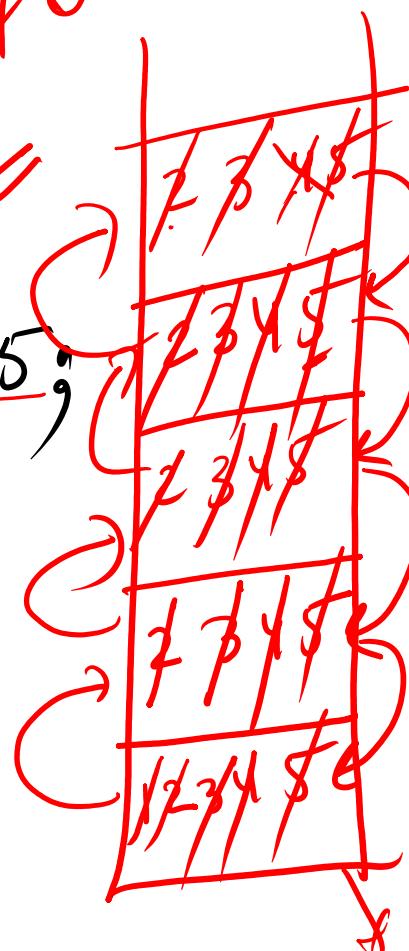
1) A function calling itself again & again,

(i) Internal Recursion(ii) External Recursion~~static~~Internal Recursion

main()

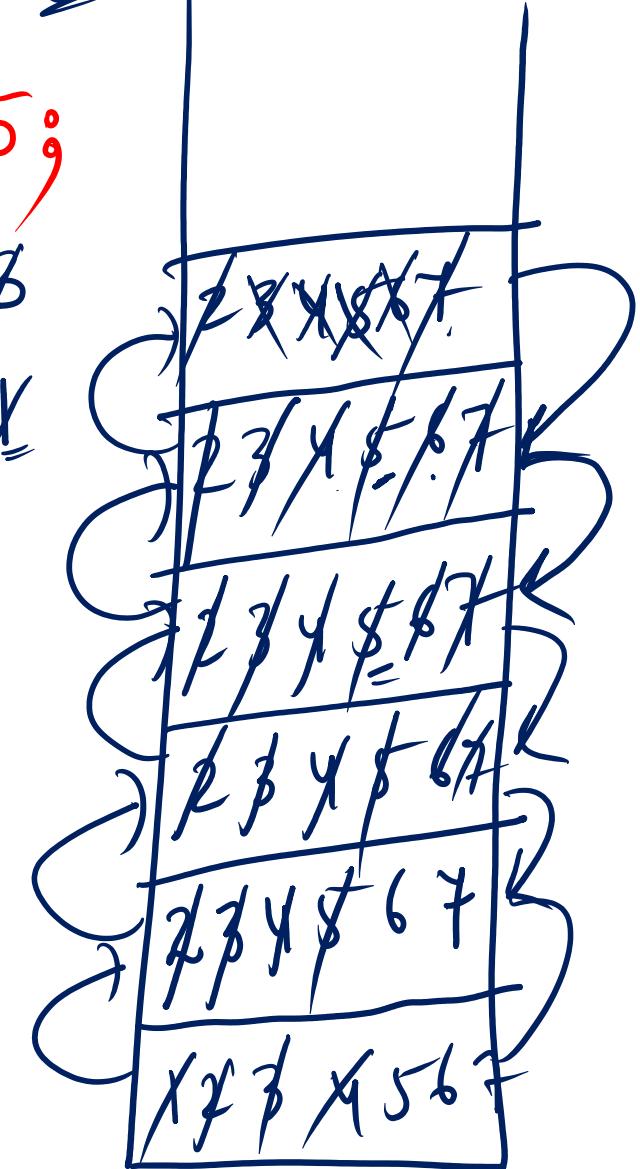
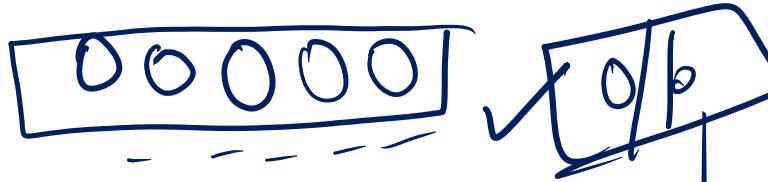
1 ~~int static int~~  $i = 5$ ;2 ~~pf(i--);~~3 ~~if (i)~~4 ~~main();~~5 ~~}~~ $i = 543210$ 

5 4 3 2 1



Examples.2

```
main()
{
    static int var = 5;
    if (var)
    {
        var--;
    }
    main();
    pf(var);
}
```





## Examples.2

Prep Smart. Score Better.

## ✓ External Recursion

```

main()
{
    int n=5, fact;
    fact=factorial (n);
    pf(fact);
}

```

5  
5  
5

$n = 5 * 1 2 3 4$   
 $n = 5 * \text{fact}(n-1)$   
 $1 2 3 4$

```

int factorial (int n)
{
    if (n > 1)
        return n * factorial (n--);
    else
        return n;
}

```

5  
-n  
n \* factorial (n--);  
post

infinite loop

n-5  
1 2 3 4  
 $n = 5 * \text{fact}(n-1)$   
1 2 3 4

$(n-1)^4$   
n \* f(n-1)  
 $n = 5^4$

(UGCNET-june2008-ii-13) Consider the following C code:

```
{ int a=5, b=9;  
float r; }  
r=b/a; }  
  
g/5 = 1  
1.0
```

What is the value of r ?

(A) 1.8

✓ (B) 1.0

(C) 2.0

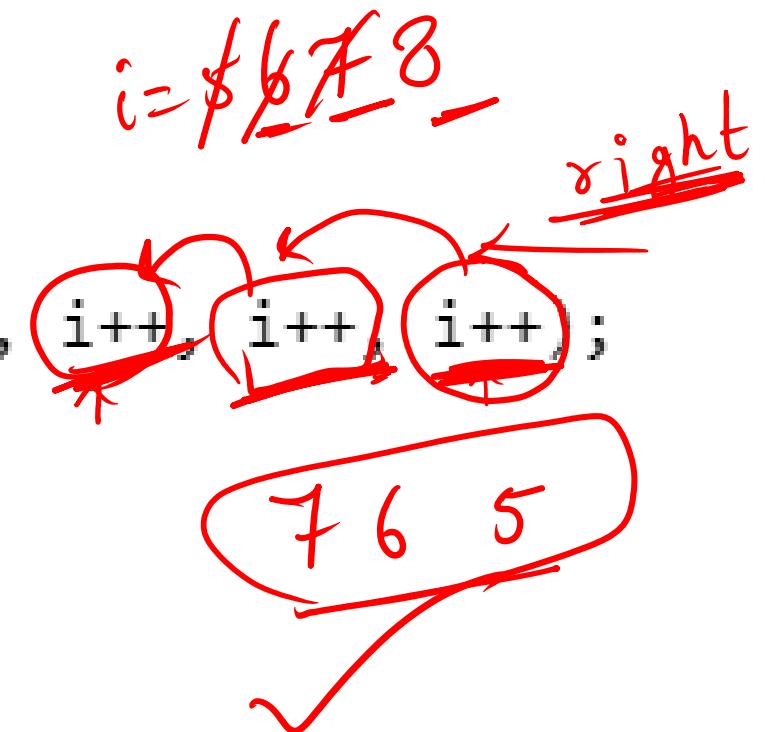
(D) 0.0



## Output of following program?

```
#include <stdio.h>
int main()
{
    int i = 5;
    printf("%d %d %d", i++, i++, i++);
    return 0;
}
```

Compiler dependent



7 6 5

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("%d", main);
```

```
    return 0;
```

```
}
```

base Address



Address of main function

Compiler Error X

Runtime Error X

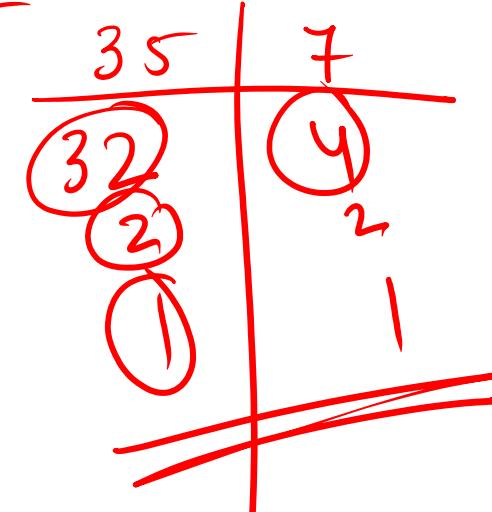
Some random value X

AND  $\rightarrow$   $\ll$  \*

(\*)

The bitwise OR of 35 with 7 in C will be:

- (A) 35 (B) 7  
(C) 42 (D) 39



(\*) What will be the output of the following 'C' code  
?

```
main ()
{ int x = 128;
printf ("\n%d", 1 + x++);
}
```

(A) 128 (B) 129  
(C) 130 (D) 131

$x = 128\ 129$

$1 + 128$   
129

(B)