

Q.35 #include<stdio.h>

```

int fun1()
{
    printf("GATE ACADEMY");
    return 0;
}

int fun2()
{
    printf("Pankaj");
    return 0;
}

int main()
{
    int (*p[2])();
    p[0] = fun1;
    p[1] = fun2;
    p[1]();
    return 0;
}

```

(A) Pankaj

(C) Compilation Error

(B) GATE ACADEMY

(D) pankajGATEACADEMY

Q.36 #include<stdio.h>

GATE 2018

```

void fun1(char *s1, char *s2) {
    char *tmp;
    tmp = s1;
    s1 = s2;
    s2 = tmp;
}

void fun2(char **s2, char **s2) {
    char *tmp;
    tmp = *s1;
    *s1 = *s2;
    *s2 = tmp;
}

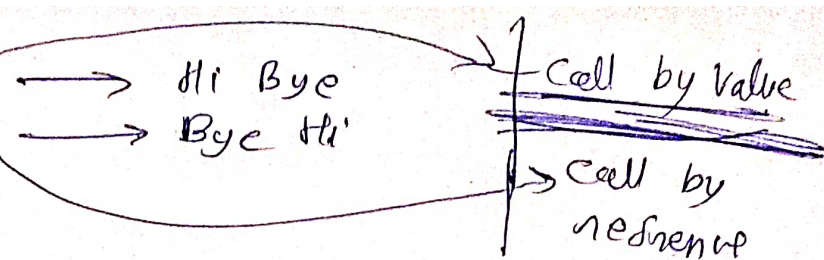
int main () {

```

```

char *str1 = "Hi", *str2 = "Bye";
fun1(str1, str2); printf("%s %s", str1, str2);
fun2(&str1, &str2); printf("%s %s", str1, str2);
return 0;
}

```



The output of the program above is

- (A) Hi Bye Bye Hi
 (B) Hi Bye Hi Bye
 (C) Bye Hi Hi Bye
 (D) Bye Hi ByeHi

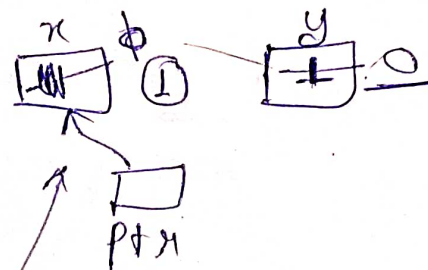
Q.37 Consider the following function implemented in C:

GATE -2017

```

void printxy (int x, int y) {
    int *ptr;
    x = 0;
    ptr = &x;
    y = *ptr;
    *ptr = 1;
    printf("%d,%d",x,y);
}

```



The output of invoking printxy(1, 1) is

- (A) 0, 0
 (B) 0, 1
 (C) 1, 0
 (D) 1, 1

Q.38 #include <stdio.h>

GATE CS-2003

```

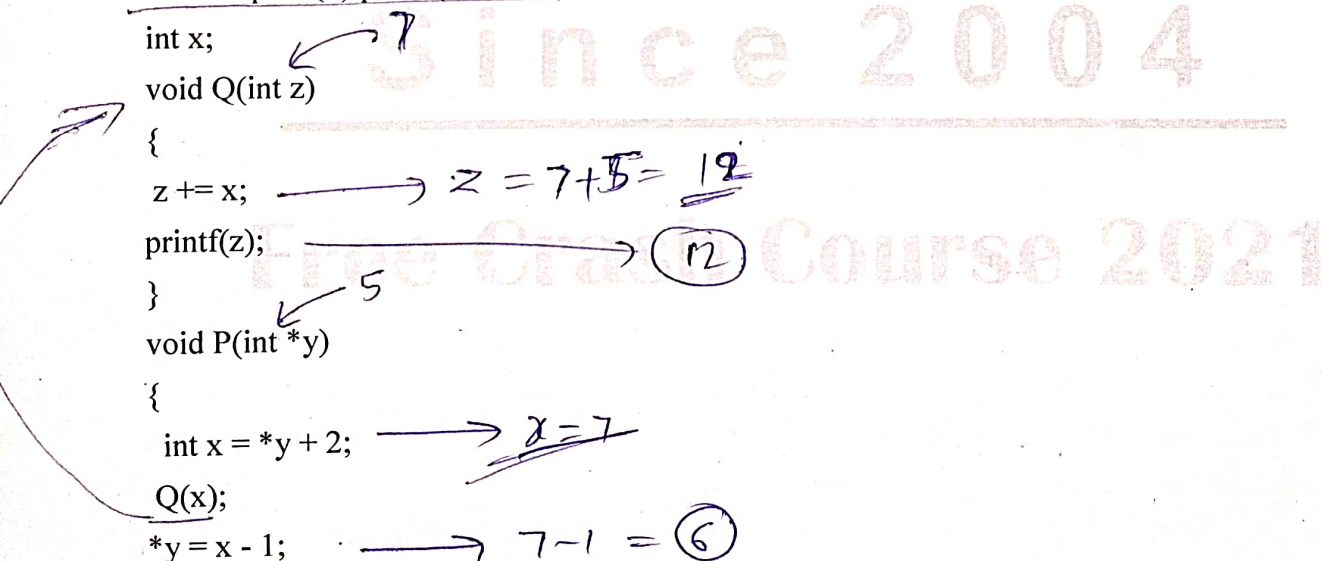
#define print(x) printf("%d ", x)

```

```

int x;
void Q(int z)
{
    z += x;
    printf(z);
}
void P(int *y)
{
    int x = *y + 2;
    Q(x);
    *y = x - 1;
}

```



```
printf(x);
```

```
}
```

```
main(void)
```

```
{
```

```
    x = 5;
```

```
    P(&x);
```

```
    printf(x);
```

```
}
```

The output of this program is

(A) 12 7 6

(B) 22 12 11

(C) 14 6 6

(D) 7 6 6

Q.39 What is printed by the following C program?

GATE-2008

```
int f(int x, int *py, int **ppz)
```

```
{
```

```
    int y, z;
```

```
    **ppz += 1;
```

```
    z = **ppz;
```

```
    *py += 2;
```

```
    y = *py;
```

```
    x += 3;
```

```
    return x + y + z;
```

```
}
```

```
void main()
```

```
{
```

```
    int c, *b, **a;
```

```
    c = 4;
```

```
    b = &c;
```

```
    a = &b;
```

```
    printf( "%d", f(c,b,a));
```

```
    getchar();
```

```
}
```

(A) 18

(B) 19

(C) 21

(D) 22

Q.40 What does the following program print ?

GATE 2010

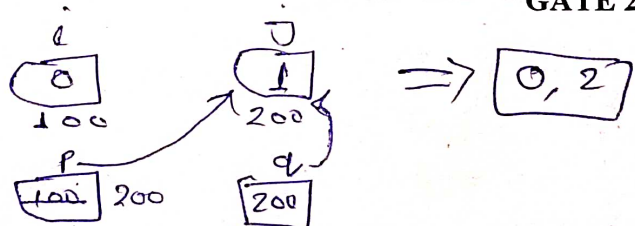
```
#include<stdio.h>
```

```
void f(int *p, int *q){
```

```
    p=q;
```

```
    *p=2;
```

```
}
```




```

int i=0,j=1; // Global
int main(){
f(&i,&j);
printf("%d %d\n",i,j);
return 0;
}

```

(A) 2 2

(B) 2 1

(C) 0 1

(D) 0 2

Q.41 Consider the following recursive C function that takes two arguments unsigned int foo(unsigned int n, unsigned int r) {

```

if (n > 0) return (n%r + foo (n/r, r));
else return 0;
}

```

$$\rightarrow 345 \div 10 = 34 \text{ remainder } 5$$

$$34 \div 10 = 3 \text{ remainder } 4$$

$$3 \div 10 = 0 \text{ remainder } 3$$

$$5 + 4 + 3 = 12$$

What is the return value of the function foo when it is called as foo(345, 10) ?

(A) 345

(B) 12

(C) 5

(D) 3

GATE -2011

Q.42 Find the output of the following code :

```

#include<stdio.h>

```

```

void Fun(int);

```

```

int main()
{

```

```

Fun(3);

```

```

return 0;
}

```

```

void Fun(int n){

```

```

if(n>0){

```

```

printf("%d",n);

```

```

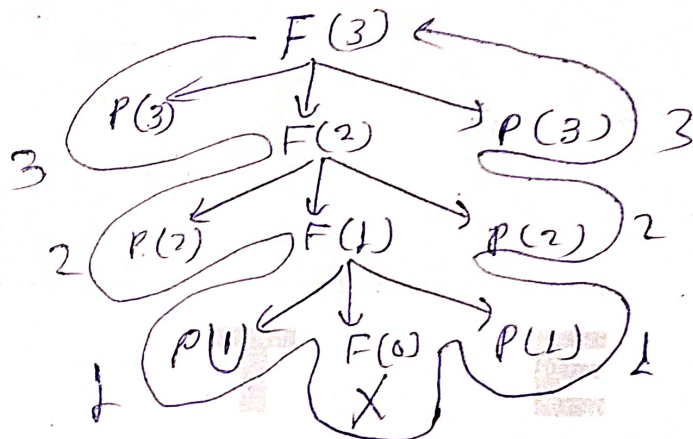
Fun(n-1);

```

```

printf("%d",n);
}
}

```



(A) 332211

(B) 321321

(C) 321123

(D) None of these

Q.43 Find the output of the following code :

```

#include<stdio.h>

```

```

void Fun(int);

```

```

int main()
{

```

```

{

```

```

Fun(3);
}
}

```

```
return 0;
```

```
}  
void Fun(int n){  
if(n>0){
```

```
Fun(n-1);  
printf("%d ",n);  
Fun(n-1);  
}
```

(A) 1211213

(B) 121

(C) 1213121

(D) 121121

Q.44 Find the output of the following code

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

```
int Fun(int);
```

```
int main()
```

```
{  
printf("%d",Fun(6));
```

```
return 0;
```

```
}  
int Fun(int n){
```

```
if(n<=1)
```

```
return n;
```

```
return Fun(n/2) + Fun(n/2) + 1;
```

```
}
```

(A) 3

(B) 6

(C) 7

(D) 4

Q.45 Find the output of the following code

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

```
int Fun(int);
```

```
int main()
```

```
{  
printf("%d",Fun(6));
```

```
return 0;
```

```
}
```

```
int Fun(int n){
```

```
if(n<=1)
```

```
return n;
```

```
return 2* Fun(n/2) + 1;
```

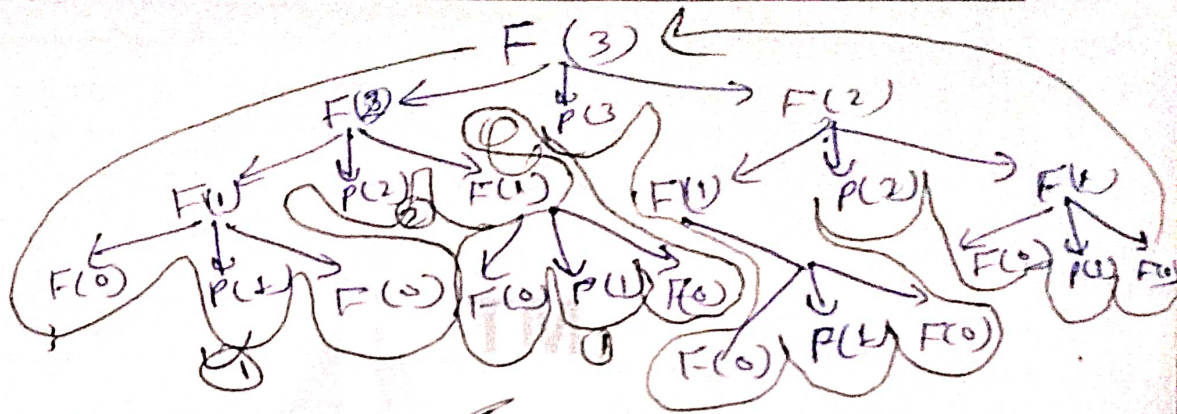
```
}
```

(A) 3

(B) 6

(C) 7

(D) 4



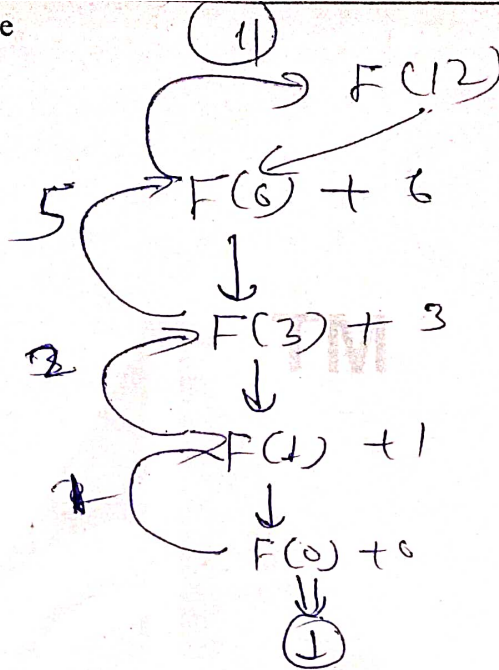
$$\begin{aligned} & \text{Fun}(6) \\ & 2 * \text{Fun}(3) + 1 \Rightarrow 2 * 3 + 1 = 7 \\ & 2 * \text{Fun}(1) + 1 \Rightarrow 2 * 1 + 1 = 3 \end{aligned}$$

Since 2004

Free Crash Course 2021

Q.46 Find the output of the following code

```
#include<stdio.h>
int Fun(int);
int main()
{
printf("%d",Fun(12));
return 0;
}
int Fun(int n){
if(n<=1)
return n;
return Fun(n/2) + n/2;
}
```



(A) 13

(B) 12

(C) 7

(D) 11

Q.47 void foo(int n, int sum)

```
{
int k = 0, j = 0;
if (n == 0) return;
k = n % 10;
j = n / 10;
sum = sum + k;
foo(j, sum);
printf("%d,", k);
}
```

int main ()

```
{
int a = 2048, sum = 0;
foo (a, sum);
printf ("%d\n", sum);
getchar();
}
```

(2048, 0)

8, 4, 0, 2
204, 20, 2, 0
8, 12, 12, 14

2, 0, 4, 8

0 ("call by value")

GATE 2005

What does the above program print?

(A) 8, 4, 0, 2, 14

(B) 8, 4, 0, 2, 0

(C) 2, 0, 4, 8, 14

(D) 2, 0, 4, 8, 0

Q.48 Consider the following function written in the C programming language.

GATE 2015

```
void foo (char *a)
{
    if (*a && *a != '\0')
    {
        foo (a+1);
        putchar (*a);
    }
}
```

The output of the above function on input "ABCD EFGH" is

- (A) ABCD EFGH
(B) ABCD
(C) HGFE DCBA
(D) DCBA

Q.49 Consider the following C function.

```
int fun(int n) {
int x=1, k;
  if (n==1) return x;
  for (k=1; k<n; ++k)
    x = x + fun(k) * fun (n-k);
  return x;
}
```

The return value of fun(5) is_____.

- (A) 51 (B) 52 (C) 53 (D) 54

Q.50 Consider the following recursive C function. If get(6) function is being called in main() then how many times will the get() function be invoked before returning to the main()? GATE-2015

GATE-2015

```
void get (int n)
```

```
{
    if (n < 1) return;
    get(n-1);
    get(n-3);
    printf("%d", n);
}
```

- (A) 15 (B) 25 (C) 35 (D) 45

Q.51 What is the value printed by the following C program ?

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

```
int f(int *a, int n)
```

{

```
if (n<=0)
```

GATE-2010

```

func1(y,x,x);
print x;
print y;
}

```

```

func1(x,y,z)
{

```

$y = y + 4;$

$z = x + y + z;$

(A) 10, 3

(C) 27, 7

(B) 31, 3

(D) None of the above

Q.54 Assume the following C variable declaration:

`int *A[10], B[10][10];`

GATE -2003

Of the following expressions:

I. `A[2]` ————— take pointer

II. `A[2][3]` ————— take int

III. `B[1]`

IV. `B[2][3]` ————— take int

Base Address of B

Which will not give compile-time errors if used as left hand sides of assignment statements in a C program?

(A) I, II, and IV only

(B) II, III, and IV only

(C) II and IV only

(D) IV only

Q.55 What is the output of the following program?

GATE 2004

`#include <stdio.h>`

`int funcf (int x);`

`int funcg (int y);`

`main ()`

`{`

`int x = 5, y = 10, count;`

`for (count = 1; count <= 2; ++count) {`

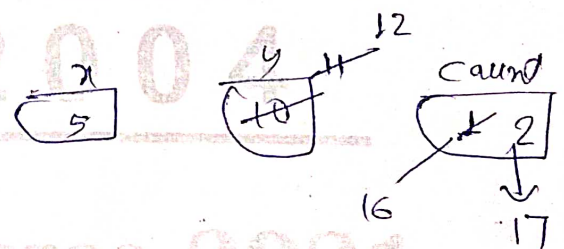
`y += funcf(x) + funcg(x);`

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

`}`

`}`

`funcf (int x) {`



$10 + 16 + 17 =$

$43 + 18 + 19 =$

43
80

Ans


```

int y;
y = funcg(x);
return (y);
}
funcg (int x) {
    static int y = 10;
    y += 1;
    return (y + x);
}

```

Handwritten annotations: An arrow points from the `return (y);` line to a circled `16`. Another arrow points from the closing brace of `funcg` to a circled `5`. A third arrow points from the `y += 1;` line to a handwritten `11`. A fourth arrow points from the `return (y + x);` line to a handwritten calculation `11 + 5 = 16`, which is then circled.

(A) 43 80

(C) 33 37

(B) 42 74

(D) 32 32