C GATE

Consider the following C function:

```
int f(int n)
{
static int i = 1;
if (n >= 5)
return n;
n = n+i;
i++;
return f(n);
}
```

The value returned by f(1) is (GATE CS 2004)

- ☐ 5
- ☐ 6
- ☐ *7*
- ☐ 8

Since i is static, first line of f() is executed only once.

Choose the correct option to fill ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a newline character.

```
void reverse(void)
{
int c;
if (?1) reverse() ;
?2
}
main()
{
printf ("Enter Text ") ;
printf ("\n") ;
reverse();
printf ("\n") ;
}
```

GATE 2008

- ☐ ?1 is (getchar() != ’\n’) ?2 is getchar(c);
- ☐ ?1 is (c = getchar()) != ’\n’) ?2 is getchar(c);
- ☐ ?1 is (c != ’\n’) ?2 is putchar(c);
- ☐ ?1 is ((c = getchar()) != ’\n’) ?2 is putchar(c);

The following C declarations

```
struct node
{
int i;
float j;
};
struct node *s[10] ;
```
define s to be (GATE CS 2000)

- ☐ An array, each element of which is a pointer to a structure of type node
- ☐ A structure of 2 fields, each field being a pointer to an array of 10 elements
- ☐ A structure of 3 fields: an integer, a float, and an array of 10 elements
- ☐ An array, each element of which is a structure of type node.

What is the value printed by the following C program?
```
#include
int f(int *a, int n)
{
if(n <= 0) return 0;
else if(*a % 2 == 0) return *a + f(a+1, n-1);
else return *a - f(a+1, n-1);
}
int main()
{
int a[] = {12, 7, 13, 4, 11, 6};
printf("%d", f(a, 6));
getchar();
return 0;
}
```
GATE 2010

- ☐ -9
- ☐ 5
- ☐ **15**
- ☐ 19

What will be the output of the following C program segment?
```
char inchar = \\\'A\\\';
switch (inchar)
{
case \\\'A\\\' :
printf ("choice A \\\\n") ;
case \\\'B\\\' :
printf ("choice B ") ;
case \\\'C\\\' :
case \\\'D\\\' :
case \\\'E\\\' :
default:
printf ("No Choice") ;
}
```

- ☐ No choice

- ☐ Choice A
- ☐ Choice A Choice B No choice
- ☐ Program gives no output as it is erroneous

Consider this C code to swap two integers and these five statements: the code

```
void swap(int *px, int *py)
{
*px = *px - *py;
*py = *px + *py;
*px = *py - *px;
}
```

S1: will generate a compilation error
S2: may generate a segmentation fault at runtime depending on the arguments passed
S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process
S4: implements the swap procedure correctly for some but not all valid input pointers
S5 :may add or subtract integers and pointers
GATE 2006

- ☐ S1
- ☐ S2 and S3
- ☐ S2 and S4
- ☐ S2 and S5

Consider the following C-function in which a[n] and b[m] are two sorted integer arrays and c[n + m] be another integer array.

```
void xyz(int a[], int b [], int c[])
{
int i, j, k;
i = j = k = O;
while ((i<m))
 if (a[i] < b[j]) c[k++] = a[i++];
else c[k++] = b[j++];
}
```

Which of the following condition(s) hold(s) after the termination of the while loop?
(i) j < m, k = n+j-1, and a[n-1] < b[j] if i = n
(ii) i < n, k = m+i-1, and b[
GATE 2006

- ☐ only (i)
- ☐ only (ii)
- ☐ either (i) or (ii) but not both
- ☐ neither (i) nor (ii)

Consider the C program shown below.

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

```
int x;
void Q(int z)
{
z += x;
print(z);
}
void P(int *y)
{
int x = *y+2;
Q(x);
*y = x-1;
print(x);
}
main(void)
{ x=5;
P(&x);
print(x);
getchar();
}
```

The output of this program is (GATE CS 2003)

- ☐ 1276
- ☐ 22 12 11
- ☐ 14 6 6
- ☐ 766

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;
}
```

GATE 2011

- ☐ 9
- ☐ 8
- ☐ 5
- ☐ 2

he number of tokens in the following C statement.

```
printf("i = %d, &i = %x", i, &i);
```

is (GATE 2000)

- ☐ 3
- ☐ 26
- ☐ 10

- ☐ 21

  What is printed by the following C program?

  ```
  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();
  }
  ```

  GATE 2008
- ☐ 18
- ☐ 19
- ☐ 21
- ☐ 22

  Consider the following C function

  ```
  int f1(int n)
  {
  if(n == 0 || n == 1)
  return n;
  else
  return (2*f1(n-1) + 3*f1(n-2));
  }
  int f2(int n)
  {
  int i;
  int X[N], Y[N], Z[N] ;
  X[0] = Y[0] = Z[0] = 0;
  X[1] = 1; Y[1] = 2; Z[1] = 3;
  for(i = 2; i <= n; i++)
  {
  X[i] = Y[i-1] + Z[i-2];
  ```

```
Y[i] = 2*X[i];
Z[i] = 3*X[i];
}
return X[n] ;
}
```

The running time of f1(n) and f2(n)
GATE 2008

- ☐ theta(n) and theta (n)
- ☐ theta(2^n) and theta(n)
- ☐ theta(n) and theta(2^n)
- ☐ theta(2^n) and theta(2^n)

Consider the following three C functions :,

```
[PI] int * g (void)
{
int x = 10;
return (&x);
}
[P2] int * g (void)
{
int * px;
*px = 10;
return px;
}
[P3] int *g (void)
{
int *px;
px = (int *) malloc (sizeof(int));
*px = 10;
return px;
}
```

Which of the above three functions are likely to cause problems with pointers? (GATE 2001)

- ☐ Only P3
- ☐ Only P1 and P3
- ☐ Only P1 and P2
- ☐ P1, P2 and P3

In the C language (GATE CS 2002)

- ☐ At most one activation record exists between the current activation record and the activation record for the main
- ☐ The number of activation records between the current activation record and the activation record for the main depends on the actual function calling sequence.
- ☐ he visibility of global variables depends on the actual function calling sequence.

- ☐ Recursion requires the activation record for the recursive function to be saved on a different stack before the recursive function can be called.

  Consider the following C declaration

```
struct {
short s [5]
union {
float y;
long z;
}u;
} t;
```

  Assume that objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment considerations, is (GATE CS 2000)

- ☐ 22 bytes
- ☐ 14 bytes
- ☐ 18 bytes
- ☐ 10 bytes

  Consider the following C function

```
void swap (int a, int b)
{
int temp;
temp = a;
a = b;
b = temp;
}
```

  In order to exchange the values of two variables x and y. (GATE CS 2004)

- ☐ call swap (x, y)
- ☐ call swap (&x, &y)
- ☐ swap (x,y) cannot be used as it does not return any value
- ☐ swap (x,y) cannot be used as the parameters are passed by value

  Consider the following C program segment:

```
char p[20];
char *s = "string";
int length = strlen(s);
int i;
for (i = 0; i < length; i++)
p[i] = s[length — i];
printf("%s",p);
```

  The output of the program is (GATE CS 2004)

- ☐ gnirts
- ☐ gnirt

- ☐ string
- ☐ no output is printed

Consider the following program fragment for reversing the digits in a given integer to obtain a new integer. Let n = D1D2…Dm

```
int n, rev;
rev = 0;
while (n > 0)
{
rev = rev*10 + n%10;
n = n/10;
}
```

The loop invariant condition at the end of the ith iteration is:(GATE CS 2004)

- ☐ n = D1D2….Dm-i and rev = DmDm-1…Dm-i+1
- ☐ n = Dm-i+1…Dm-1Dm and rev = Dm-1….D2D1
- ☐ n =! rev
- ☐ n = D1D2….Dm and rev = DmDm-1…D2D1

What does the following program print?

```
#include
void f(int *p, int *q)
{
p = q;
*p = 2;
}
int i = 0, j = 1;
int main()
{
f(&i, &j);
printf("%d %d \n", i, j);
getchar();
return 0;
}
```

GATE 2011

- ☐ 2 2
- ☐ 2 1
- ☐ 0 1
- ☐ 0 2

Consider the following C programint a, b, c = 0;

```
void prtFun (void);
int main ()
{
static int a = 1; /* line 1 */
```

```
prtFun();
a += 1;
prtFun();
printf ( "\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n %d %d " , a, b) ;
}
void prtFun (void)
{
static int a = 2; /* line 2 */
int b = 1;
a += ++b;
printf (" \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n %d %
```
GATE 2012

- ☐ 3 1 4 1 4 2
- ☐ 4 2 6 1 6 1
- ☐ 4 2 6 2 2 0
- ☐ 3 1 5 2 5 2

What does the following fragment of C-program print?
```
char c[] = "GATE2011";
char *p =c;
printf("%s", p + p[3] - p[1]) ;
```

GATE 2011

- ☐ GATE2011
- ☐ E2011
- ☐ 2011
- ☐ 011

Consider the following C program
```
main()
{
int x, y, m, n;
scanf ("%d %d", &x, &y);
/* x > 0 and y > 0 */
m = x; n = y;
while (m != n)
{
if(m>n)
m = m - n;
else
n = n - m;
}
printf("%d", n);
}
```
The program computes (GATE CS 2004)

- ☐ x + y using repeated subtraction
- ☐ x mod y using repeated subtraction
- ☐ the greatest common divisor of x and y
- ☐ the least common multiple of x and y

Consider the following three C functions :,

```
[PI] int * g (void)
{
int x = 10;
return (&x);
}
[P2] int * g (void)
{
int * px;
*px = 10;
return px;
}
[P3] int *g (void)
{
int *px;
px = (int *) malloc (sizeof(int));
*px = 10;
return px;
}
```

Which of the above three functions are likely to cause problems with pointers? (GATE 2001)

- ☐ Only P3
- ☐ Only P1 and P3
- ☐ Only P1 and P2
- ☐ P1, P2 and P3

The C language is. (GATE CS 2002)

- ☐ A context free language
- ☐ A context sensitive language
- ☐ A regular language
- ☐ Parsable fully only by a Turing machine

The value of j at the end of the execution of the following C program. (GATE CS 2000)

```
int incr (int i)
{
static int count = 0;
count = count + i;
return (count);
}
main ()
```

```
{
int i,j;
for (i = 0; i <=4; i++)
j = incr(i);
}
```

- ☐ 10
- ☐ 4
- ☐ 6
- ☐ 7

Assume the following C variable declaration
int *A [10], B[10][10];
Of the following expressions
I A[2]
II A[2][3]
III B[1]
IV B[2][3]

which will not give compile-time errors if used as left hand sides of assignment statements in a C program (GATE CS 2003)?

- ☐ I, II, and IV only
- ☐ II, III, and IV only
- ☐ II and IV only
- ☐ IV only

Consider the following C-program fragment in which i, j and n are integer variables.
for (i = n, j = 0; i >0; i /= 2, j += i);
Let val(j) denote the value stored in the variable j after termination of the for loop. Which one of the following is true?
GATE 2006

- ☐ val(j) = theta(logn)
- ☐ vaI(j) = theta (sqrt(n))
- ☐ val(j) = theta(n)
- ☐ val(j) = theta(nlogn)

Consider the following declaration of a ‘two-dimensional array in C:
char a[100][100];

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of a[40][50] is (GATE CS 2002)

- ☐ 4040
- ☐ 4050

- ☐ 5040
- ☐ 5050

Submit

Read more: http://www.techvyom.com/c-programming-solved-questions-from-previous-years-gate-papers.html#ixzz3jSgkwzhj

```
#include<stdio.h>

void f(int *p, int *q)

{

  p = q;

 *p = 2;

}

int i = 0, j = 1;

int main()

{

  f(&i, &j);

  printf("%d %d \n", i, j);

  getchar();
  return 0;

}
```

Run on IDE

(A) 2 2
(B) 2 1
(C) 0 1
(D) 0 2

What does the following fragment of C-program print?

```
char c[] = "GATE2011";

char *p =c;

printf("%s", p + p[3] - p[1]) ;
```

Run on IDE

(A) GATE2011
(B) E2011
(C) 2011
(D) 011

```
int a, b, c = 0;

void prtFun (void);

int main ()

{

    static int a = 1; /* line 1 */

    prtFun();
    a += 1;

    prtFun();

    printf ( "\n %d %d " , a, b) ;

}



void prtFun (void)

{

    static int a = 2; /* line 2 */

    int b = 1;

    a += ++b;

    printf (" \n %d %d " , a, b);

}
```

Run on IDE

What output will be generated by the given code segment?
(A) 3 1

4 1
4 2
(B) 4 2
6 1
6 1
(C) 4 2
6 2
2 0

(D) 3 1
5 2
5 2

What output will be generated by the given code d\segment if:
Line 1 is replaced by “auto int a = 1;”
Line 2 is replaced by “register int a = 2;”

A) 3 1
4 1
4 2
(B) 4 2
6 1
6 1
(C) 4 2
6 2
2 0
(D) 4 2
4 2
2 0

Consider the following C function

```
void swap (int & a, int &b)
{
int temp;
temp = a;
a = b;
b = temp;
}
```

In order to exchange the values of two variables x and y.
call swap(&x,&y)   call swap (x,y)   call swap(*x,*y)   swap (x,y) cannot be used as the parameters are passed by value

Which of the following is a valid C/C++ function pointer definition

int (*f)();   int* f(); (int*)f();   (int* f)();

If a binary operator is overloaded as a non-member function, how many parameters will the non member function required?

None. Both operands are already known

One, to pass the first operand. The second operand will be a member of the first.

Binary operators can't be overloaded as global functions, only as member functions of a class.

Two, to pass the first and second operands.

Which of these is not true about static data member?

Can be accessed only static members

Each object of the class will have its own copy of static data member

Memory will be allocates only once during the class declaration

When any object modifies static data member , the result will be visible to all instances of the class

Which of these is false about static member functions?

A Static member functions cannot be defined in the private section of a class

B Can access only static data members of the class

C Static member functions can invoke other static member functions

D All above are false

The mechanism that binds code and data together to keep them secure from outside world is known as
Abstraction Encapsulation Inheritance Polymorphism

What will be output if you will compile and execute the following c code?

```
#define x 5+2

void main(){

   int i;

   i=x*x*x;

   printf("%d",i);

}
```

343 27 34 29

```
#include<stdio.h>
int main(){
int i = 5 , j;
int *p , *q;
p = &i;
q = &j;
j = 5;
printf("%d %d",*p,*q);
```

```
 return 0;
```
5 5   Address Address   Complitation error   none of the above

Consider the following C program segment.

```
# include <stdio.h>
int main()
{
    char s1[7] = "1234", *p;
    p = s1 + 2;
    *p = '0';
    printf("%s", s1);
}
```

What will be printed by the program?

12   120400 1204   1034