Practice Session - IV (GATE PYQs)

Sunday, 30 June 2024 8:19 PM

1.

GATE CS, 10

What does the following program print?

```
#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);
    return 0;
}
```

a) 2 2

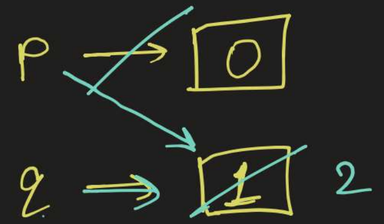
b) 2 1

c) 0 1

~~d) 0 2~~

Sol =

i = 0 , j = 1



i = 0

j = 2

(2.)

GATE CS, 11.

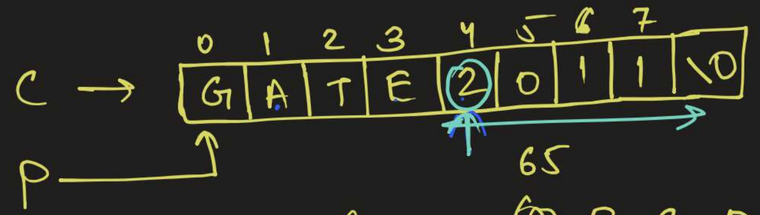
What does the following fragment of C program print?

```
char c[] = "GATE2011";
char *p = c;
printf("%s", p + p[3] - p[1]);
```

4 => printf("%s", p+4);

a.) GATE2011
b.) E2011
c.) 2011
d.) 011

Sol.



65 69
A, B, C, D, E

P[3] = 'E'
P[1] = 'A'

P[3] - P[1] = 4.

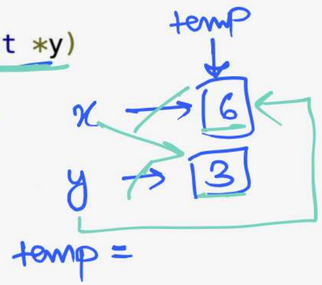
P+4 → 2011.

3.

GATE CS, 06

Which one of the choices given below would be printed when the following program is executed?

```c
#include <stdio.h>
void swap (int *x, int *y)
{
    static int *temp;
    temp = x;
    x = y;
    y = temp;
}
void printab ()
{
    static int a = -3, b = -6;
    int i = 0;
    while (i <= 4)
    {
        if ((i++)%2 == 1) continue;
        a = a + i;
        b = b + i;
    }
    swap (&a, &b);
    printf("a =  %d, b = %d\n", a, b);
}
main()
{
    printab();
    printab();
}
```



a) a=0, b=3
   a=0, b=3

b) a=3, b=0
   a=12, b=9

c) a=3, b=6
   a=3, b=4

d) a=6, b=3
   a=15, b=12

0, 2, 4 | 1, 3, & 5

⇒ a = a+1 = -3+1+3+5

⇒ b = -6+1+3+5

a=6, b=3.

I	II
a = 6	a = 6 + 9 = 15
b = 3	b = 3 + 9 = 12

Sol. i = 0, 0%2 = 0 == 1 → false.

a = a + 1

b = b + 1

i = 0, 2, 4

= 1, 3, 5 } (3)

4.

GATE CS, 07.

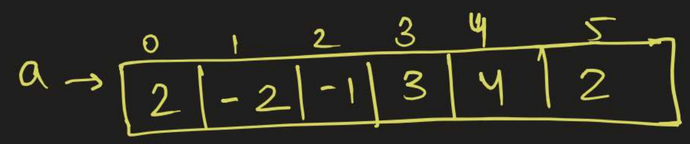
Consider the C program given below :

```c
#include <stdio.h>
int main ()    {
    int sum = 0, maxsum = 0,  i,  n = 6;
    int a [] = {2, -2, -1, 3, 4, 2};
    for (i = 0; i < n; i++)    {
            if (i == 0 || a [i]  < 0  || a [i] < a [i - 1])  {
                    if (sum > maxsum) maxsum = sum;
                    sum = (a [i] > 0) ? a [i] : 0;
            }
            else sum += a [i];
    }
    if (sum > maxsum) maxsum = sum ;
    printf ("%d\n", maxsum);

}
```

a) 9
b) 8
c) 7
d) 6

Sol =



i = 0 ; a[0] = 2 , m = 0, S = 2
i = 1 ; a[1] = -2 , m = 2, S = 0
i = 2 ; a[2] = -1 , m = 2, S = 0
i = 3 ; a[3] = 3 , m = 2, S = 3
i = 4 ; a[4] = 4 , m = 2, S = 7
i = 5 ; a[5] = 2 , m = 7, S = 2

maxSum.

(5.)

GATE CS, 08

C program is given below:

```
# include <stdio.h>
int main ()
{
        int i, j;
        char a [2] [3] = {{'a', 'b', 'c'}, {'d', 'e', 'f'}};
        char b [3] [2];
        char *p = *b;
        for (i = 0; i < 2; i++) {
              for (j = 0; j < 3; j++) {
              *(p + 2*j + i) = a [i] [j];
              }
        }
}
```

Q. what will be the values of b ?

a) a b
   c d
   e f

b) a d
   b e
   c f

c) a c
   e b
   d f

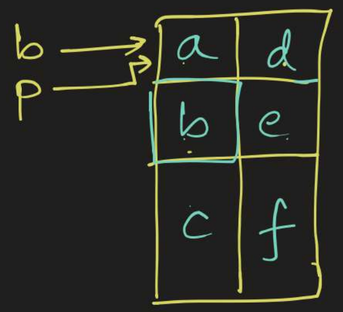
d) a e
   d c
   b f

Sol.

a →

a	b	c
d	e	f

i	j	a[i][j]	2*j+i
0	0	a	0
0	1	b	2
0	2	c	4
1	0	d	1
1	1	e	3
1	2	f	5



(6.)

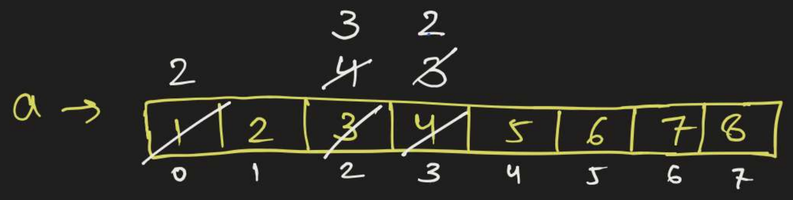
GATE CS, 08

Consider the C program given below. What does it print?

```c
#include <stdio.h>
int main ()
{
        int i, j;
        int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
        for(i = 0; i < 3; i++) {
             a[i] = a[i] + 1;
             i++;
        }
        i--;
        for (j = 7; j > 4; j--) {
             int i = j/2;
             a[i] = a[i] - 1;
        }
        printf ("%d, %d", i, a[i]);
}
```

a.) 2,2
b.) 2,4
c.) 3,2
d.) 3,3

Sol =



i = 0, 1, 2, 3
i = 2,

j = 7, i = 3,
j = 6, i = 3
j = 5, i = 2
~~j = 4~~

(7)

GATE CS,08

Consider the following C code segment.

```
int a, b, c = 0;
void prtFun(void);
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);
}
```

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

(6)

GATE CS, 14

For a C program accessing **X[i][j][k]**, the following intermediate code is generated by a compiler. Assume that the size of an **integer** is 32 bits and the size of a **character** is 8 bits.

```
t0 = i * 1024
t1 = j * 32
t2 = k * 4
t3 = t1 + t0
t4 = t3 + t2
t5 = X[t4]
```

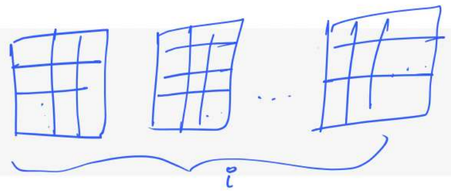
4 bytes

$t_2 = k$

1 byte

$j \times k$

char x [ ] [1024] [32]



Which one of the following statements about the source code for the C program is CORRECT?

x is declared as :

a) int x [32] [32] [8]

b) int x [4] [1024] [32]

c) char x [4] [32] [8]

d) char x [32] [16] [2]

int x [ ] [32] [8] ;

Sol. $x[t_4] = x[t_3 + t_2] = x(t_1 + t_0 + t_2) = x[i * 1024 + j * 32 + k * 4]$

k is multiplied by (4), k * size of (int)

k * size of (data type)

$x[\ i * \text{total rows} * \text{total columns} * \text{size of (int)} + j * \text{total col} * \text{size of (int)} + k * (\text{size of (int)})\ ]$

2-d array → int x [ ] [rowSize] (c)

3-d array → int x [ ] [colSize] [rowSize] (r, c)

Here, j is getting multiplied by 32

and, smallest individual unit (1 box of info) is taking 4 bytes.

(k) no. of columns = 32 / 4 = (8).

(j) no. of rows = 1024 / 32 = (32).

9.

GATE CS, 14

Suppose n and p are unsigned int variables in a C program. We wish to set p to ${}^nC_3$. If n is large, which one of the following statements is most likely to set p correctly?

A. $p = n * (n-1) * (n-2)/6;$ → might overflow the range of unsigned int.

B. $p = n * (n-1)/2 * (n-2)/3;$

C. $p = n * (n-1)/3 * (n-2)/2;$

D. $p = n * (n-1) * (n-2)/6.0;$

B → $\left[\frac{n*(n-1)}{2}\right] * \left(\frac{(n-2)}{3}\right]$ → $n*(n-1)$ → will always be divisible by 2. So, it's the safest option (no information is lost).

C → $\left[\frac{n*(n-1)}{3}\right] * \left[\frac{(n-2)}{2}\right]$ → $n(n-1)$ might not perfectly divisible by 3.