

May - Jun  
2010

Total No. of Questions—12]

[Total No. of Printed Pages—8+4

**[3762]-232**

**S.E. (I.T.) (I Sem.) EXAMINATION, 2010**  
**FUNDAMENTAL OF DATA STRUCTURES**  
**(2008 COURSE)**

**Time : Three Hours**

**Maximum Marks : 100**

- N.B. :—** (i) Answers to the two Sections should be written in separate answer books.
- (ii) From Section I answer Q. No. 1 or Q. No. 2, Q. No. 3 or Q. No. 4, Q. No. 5 or Q. No. 6 and answer Q. No. 7 or Q. No. 8, Q. No. 9 or Q. No. 10, Q. No. 11 or Q. No. 12 from Section II.
- (iii) Neat diagrams must be drawn wherever necessary.
- (iv) Figures to the right indicate full marks.

**SECTION I**

1. (a) Select the choice for the correct answer and write that choice : 2×4

(i) `int x=20, y=35;`

`x = ++y + y + y++;`

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

The above code snippet will print :

- (1) 109
- (2) 108
- (3) 106
- (4) 107

P.T.O.

(ii) `int x = 3;`

`int y;`

`y = x ^ 3;`

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

The above code snippet will print :

(1) 27

(2) 1

(3) 0

(4) None of the above

(iii) `#include<stdio.h>`

`#define square(x) x*x`

`main( )`

`{`

`int y;`

`y = 30/square(4);`

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

`}`

The above code snippet will print :

(1) 1

(2) 15

(3) 28

(4) 2



(iv) int reverse(int nx)

```
{  
    if(nx == 0)
```

```
        return 0;
```

```
    else
```

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

```
        nx--;
```

```
        reverse(nx);
```

```
    }
```

```
main( )
```

```
{
```

```
    int x=5;
```

```
    reverse(x);
```

```
}
```

The above code snippet will print :

(1) 5 4 3 2 1

(2) 1 2 3 4 5

(3) 5 4 3 2 1 0

(4) Program runs in an infinite loop

- (b) What is Union in C ? [2]  
(c) What is structure in C ? [2]  
(d) Illustrate the difference between Union and Structure with suitable examples. [6]

Or

2. (a) Select the choice for the correct answer and write that choice : 2×4

(i) `printf("%d", 4%3);`  
`printf("%d", 4%-3);`  
`printf("%d", -4%3);`  
`printf("%d", -4%-3);`

The above code snippet will print :

- (1) 1 1 1 1  
(2) 1 -1 -1 -1  
(3) 1 1 -1 -1  
(4) 1 -1 -1 1

(ii) `int a=30, b=40, x;`  
`x=(a != 10) && (b = 50);`  
`printf("%d", x);`

The above code snippet will print :

- (1) 1  
(2) 0  
(3) Compile Error  
(4) None of the above



(iii) `#include<stdio.h>`

`#define a 10`

`main( )`

`{`

`int a=50;`

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

`}`

The above code snippet will print :

(1) 10

(2) 50

(3) Compile error

(4) None of the above

(iv) `int main(void)`

`{`

`int x=1;`

`if(!x)`

`printf("Hello\n");`

`else`

`{`

`x=0;`

`printf("Bye\n");`

`}`

`return 0;`

`}`

The above code snippet will print :

(1) Bye

(2) Hello

(3) Hello (infinitely ....)

(4) Bye (infinitely ....)

(b) Declare Employee structure with fields as id, name and salary for 10 employees and write C code to accept and display the employee information. [6]

(c) Explain use of "break" and "continue" keywords in "C" with suitable example. [4]

3. (a) Write the output for the following code : 4×2

(i) `int array[ ] = {45, 67, 89};`

`int *array_ptr = array;`

`printf("first element :%d\n", *(array_ptr++));`

`printf("second element :%d\n", *(array_ptr++));`

`printf("third element :%d\n", *array_ptr);`

(ii) `int i, j[2]={0};`

`int *p, *q;`

`p = &i;`

`q = &j[0];`

`*p=3;`

`*++(q)=1;`

`printf("%d%d%d", i, j[0], j[1]);`



(b) Explain pass by value method of parameter passing with suitable example. [4]

(c) What will be the output of the code snippet below : [4]

```
void f1(int *px)
```

```
{
```

```
int n=100;
```

```
px=&n;
```

```
}
```

```
int main(void)
```

```
{
```

```
int *x;
```

```
f1(x);
```

```
printf("%d", *x);
```

```
}
```

(1) 0

(2) Error

(3) Garbage value

(4) None of the above

Or

4. (a) Write the output for the following code : 2×4

(1) `int t[3][2][4]={2,4,3,6,1,6,7,9,8,2,1,1,2,3,7,3,1,6,2,4,0,7,9,5};`

`printf("%d", *((*(t+2)+1)+3));`

(2) `char *cp;`

`int *ip;`

`cp=(char *)0x100;`

`ip=(int *)cp;`

`ip++;`

`cp++;`

`printf("cp=%x ip=%x", cp, ip);`

(b) Explain pass.by reference method of parameter passing with suitable example. [4]

(c) What will be the correct choice of output ? [4]

`void f1(int *px)`

`{`

`int n=100;`

`px = &n;`

`}`



```
int main(void)

{

int *x;

f1(x);

printf("%d", *x);

}
```

The above code snippet will print :

- (1) 0
- (2) Error
- (3) Garbage value
- (4) None of the above

5. (a) With respect to algorithm analysis, explain the following terms : [6]

- (i) Omega notation
- (ii) Theta notation
- (iii) Big Oh notation.

- (b) Find the frequency count (F.C.) of the above code : [6]

Explain each step.

```
double IterPow(double X, int N)
```

```
{
```

```
double Result = 1;
```

```
while (N > 0)
```

```
{
```

```
Result = Result *X;
```

```
N--;
```

```
}
```

```
return Result;
```

```
}
```

- (c) Write pseudo code to find union of two sets containing integer elements. [4]

Or

6. (a) State different ways to classify data structures with suitable examples. [8]

- (b) Write a pseudo C code for addition of two sparse matrices. [8]



## SECTION II

7. (a) Sort the following and show the status after every pass using selection sort : [10]

34, 9, 78, 65, 12, -8

- (b) Search a given number using binary search, show all passes.  
Number to be searched 6.

1, 4, 9, 13, 23, 34 [6]

Or

8. (a) Sort using quicksort the following and show all passes : [10]

34, 9, 78, 65, 12, -8

- (b) Search a given number using binary search, show all passes : [6]

Number to be searched 23.

1, 4, 9, 13, 23, 34.

9. (a) Explain the concept of row major address calculation for multidimensional array with the suitable example. [6]
- (b) Explain different ways to store polynomial in different data structures with suitable example. [10]

*Or*

10. (a) Explain the concept of column major address calculation for multi-dimensional array with suitable example. [6]
- (b) Write a pseudo C code for finding the fast transpose of a sparse matrix. Compare this with the normal transpose method. [10]
11. (a) Compare Singly Linked List with Doubly Linked List. [6]
- (b) Write a pseudo C code for inserting a node at start and at end in SLL. [10]

*Or*

12. (a) Write a pseudo C code for deletion of node from DLL from first and last position. [10]
- (b) Explain the concept of Generalized Linked List with suitable example. [6]