

CS10001 Programming and Data Structures

Class Test II

October 30, 2008 Duration: 1 hrs

Name:

Roll No:

Section:

Question 1 (6)	2 (5)	3.1 (3)	3.2 (6)	Total (20)

Roll No:

1. What is printed by the following program?

[6]

```
#include <stdio.h>

int main () {

    char s[10] = "tape" ;
    char *p[10], *temp ;
    int i;

    for (i=0; i<10; i++)
        p[i] = s+i;

    printf ("p[0] = %s, p[1] = %s\n", p[0], p[1]) ;

    temp = p[0];
    p[0] = p[2];
    p[1] = temp;

    printf ("p[0] = %s, p[1] = %s\n", p[0], p[1]) ;

    *p[4] = 'r';
    *p[5] = '\\0';
    printf ("s = %s, p[0] = %s\n", s, p[0]) ;

}
```

p[0] = tape, p[1] = ape
p[0] = pe, p[1] = tape
s = taper, p[0] = per

Roll No:

2. What is printed by the following program?

[5]

```
#include <stdio.h>

struct abc {
    int a;
    int *b;
    int c[5];
};

void foo (struct abc x, struct abc y[]) {
    x.a = 25;
    *(x.b) = 50;
    x.c[0] = 30;
    y[0] = x;
}

int main () {
    struct abc x, y[5] ;
    int n = 20;
    x.a = 5;
    x.b = &n;
    x.c[0] = 10;
    y[0] = x;
    foo (x, y) ;
    printf ("    x: %d, %d \n", x.a, x.c[0]) ;
    printf ("y[0]: %d, %d \n", y[0].a, y[0].c[0]) ;
    printf ("n=%d\n", n) ;
}
```

```
x: 5, 10
y[0]: 25, 30
n=50
```

Roll No:

3.1. Define a structure `customer` to specify data of customer in a bank. The data to be stored is: *Account number* (integer), *Name* (character string having at most 50 characters), and *Balance* in account (integer). [3]

3.2. Assume data for all the 100 customers of the bank are stored in the array :

```
struct customer bank[100];
```

The function, `transaction`, is used to perform a customer request for withdrawal or deposit to her account. Every such request is represented by the following three quantities: *Account number of the customer*, *request type* (0 for deposit and 1 for withdrawal) and *amount*.

The `transaction` function returns 0 if the transaction fails and 1 otherwise. The transaction fails only when the account balance is less than the withdrawal amount requested.

The array `bank` (defined above) is another input to the `transaction` function and is suitably updated after every request. In case of a failed transaction no change is made in the `bank` array.

The header for the function `transaction` is given below, complete the body of the function. [6]

Roll No:

```
int transaction ( int account_number, int request_type, int amount,  
struct customer bank [ 100 ] ) {
```

```
}
```