

Midsem: Computer Programming (MA511)

M.Sc. (M&C), Semester: Aug-Nov 2008-2009

Dept. of Mathematics, IIT Guwahati

Date: September 25, 2008

Time: 10:00AM - 12:00AM

Attempt all questions, Q-8, Q-9, Q-11 are presented in two column format, Ignore header files for Q-4, Q-8, Q-10, Q-11.

Full Marks: 50

1. `int p=12, *ptr= &p;` then fill in the blanks, print pointers in hexadecimal format.
`printf(“%—, %—, %—, %—, %—, %—”, ptr, &ptr, *ptr, p, &p, *ptr+1);` [3]
2. Fill in the blanks, then define appropriate C structures for the following data storing and printing of 100 students. [2+4]
`printf(“%—, %—, %—, %—”, stud[i].name, stud[i].age, stud[i].roll_no, stud[i].sex);`
3. What are the differences (*at most two*) between (*i*) *break* and *return* (*ii*) *call by value* and *call by reference* (*iii*) *pointer to integer* and *pointer to character* (*iv*) *structure* and *union* (*v*) *pointer variable* and *integer variable*? [5×1]
4. Write all possible inputs of the following C-program such that it prints *at least* (*i*) Tuesday, (*ii*) Monday, (*iii*) Friday, and (*iv*) all other days. (*v*) What would be the output for input 7 ? (*vi*) Make minimum necessary correction over the following C-program such that for input 1 print Sunday, 2 print Monday, and so on ...

```
main(){
    int pd;
    scanf(“%d”, &pd);
    switch(pd){
        case 5: printf(“Thursday”);break;
        default : printf(“wrong input”);
        case 2: printf(“Monday”);
        case 3: printf(“Tuesday”);break;
        case 4: printf(“Wednesday”);
        case 6: printf(“Friday”);break;
        case 8: printf(“Saturday”);break;
        case 1: printf(“Sunday”);
    }
} /* end of main */
```

 [6]

5. Write a program that that generates a 3digits number where each digit is generated from a random number generator and the number is divisible by three. [4]
6. Write a program that read a number from keyboard and generates a random floating point number in the range 0 - the input number. [4]

7. Write a program and **flow chart** which reads two character values. Compare them with respect to ASCII values. If the first is less than the second, print the message up. If the second is less than the first, print the message down. If the numbers are equal print the message equal. In each case print their ASCII values. [3+3]

8. What would be the output (according to printf format) of the following C-program?

```
void func(char *array, int array1[ ]){    main(){
    int i;                               int i, array_r[10]={9,8,7,6,5,4,3,2,1};
    array +=8;                           char array_s[10]="987654321";
    for(i=0; i <9; i++){                 func(array_s, array_r);
        *(array1+i) = i+1;               printf("%s\n", array_s);
        *array = 'x';                    for(i=0; i <9; i++)
    } /* end of for */                    printf("%d",array_r[i]);
}                                          } /* end of main */
```

9. What would be the output (according to printf format) of the following C-program?

```
void func(int i){                          #include<stdio.h>
    if(i==0) return;                       main(){
    printf(" %d", i);                       func(6);
    func(i/2);                               } /* end of main */
    printf(" *%d*\n", i);
}                                             [5]
```

10. Fill in the blanks, then what would be the output (according to printf format) of the following C-program ?

```
main(){
    int i;
    for(i=13; i<=23; i++){
        if(i%7) printf("%d is _____ by 7\n", i);
        else printf("%d is _____ by 7\n", i);
    }
} /* end of main */ [3]
```

11. What would be the output of the following C-program ?

```
main(){
    typedef int in;
    typedef float fl;
    in i = 8, j = 5;
    fl x = 5.5;
    in k = !(i >= 10) && j! = 6;
    in l = !(i <= 8) || j! = 6;
    in n, m = !(i <= 8) && j! = 6;
    if(!(i >= 10)&&j! = 6)
        printf("%d %d %d %d\n", i + (int)x, k, l, m);
    if(!(i <= 8)||j! = 6)
        printf("%f %d %d %d\n", i + x, k++, ++l, m - -);
    if(!(i <= 8)&&j! = 6)
        printf("%d\n", k + l + m);
    printf("%d %d %d\n", k, l, m);
} /* end of main */ [4]
```