

Indian Institute of Technology Kharagpur
Department of Computer Science & Engineering
Programming & Data Structures CS11001/CS13002 1st Year B.Tech/Dual Degree/MSc
Autumn Semester 2009 Number of Students: 600

Max. Time: 2 Hours

Max. Marks: 60

Instructions:

- I All the parts of a question must be answered together. If answers to a question are interspersed with answers to other questions, only the portion appearing first will be evaluated and the remaining parts will be ignored.
 - II Wherever you have been asked to write a function, no main function is required unless explicitly asked.
 - III The question paper is complete in all respects. No further clarifications will be given. In case you need any clarifications, you are advised to make suitable assumptions and write down the same with your answer neatly.
-

Q. 1 Answer the following. [3X4=12 Marks]

a) The following function is expected to compute the factorial of a positive number **n**. Is the given code error free? If not, then using one sentence state the reason why it will not work and also give the corrected code. [1+2]

```
long fact(int n)
{
    long k=1;
    do
        k*=fact(i--);
    while(n>0);
    return k;
}
```

Ans: No, it uses both recursion and iteration.

[Award 1 mark]

Corrected code

```
long fact(int n)
{
    long k=1;
    for(i=0;i<n;i++)
        k*=i;
    return k;
}
```

OR

```
long fact(int n)
{
    if (n<=1) return 1;
    return n*fact(n-1);
}
```

[Award 2 marks]

b) Complete the following recursive function to calculate x^n , n being a non-negative integer.

```
float find_power(float x, int n){
if(n ==   0  )                               /* 1 Mark*/
    return   1  ;                             /* 1 Mark*/
else
    return   x*find_power(x,n-1)  ;           /* 1 Mark*/
```

}

- c) What will be printed by the following program? Please note: The functions **f** and **g** and the **main** function are shown in two boxes.

```
#include <stdio.h>
int k=100;
int f(int k){
    k=k+1;
    return k;
}
int g(int b){
    return(b+k);
}
```

Functions

```
main(){
    int a,k=15;
    a=f(k);
    printf("k=%d,f(k)=%d\n",k,a);
    printf("g(k)=%d\n",g(k));
}
```

Main Function

**Ans: k=15, f(k)=16
g(k)=115**

- d) What value will the following function return when called as **recur(3)**?

```
int recur(int data){
    int k;

    k=(data>2)?(recur(data-1)-recur(data-2)):1;
    return k;
}
```

Ans: 0

Q. 2) For each of the following questions, do as directed. [12 Mrks]

- (i) Convert 45.75 (in hexa-decimal form) to the binary number system. Show the steps of your answer. [2]

Hexadecimal 4 5 . 7 5
Binary 0100'0101'.0111'0101

N.B. Apostrophes are used to delineate the nibbles of hexa-decimal number in the binary representation. In actual answer they may not be shown. No marks, if no step shown. If the number is converted to decimal and then to binary, full marks may be awarded if the result is correct.

- (ii) Convert 101.125 (in decimal system) to the hexadecimal system. Show the steps of your answer. [2]

Decimal to Binary:

101 → 1100101 and .125 → .001

Binary: 0110' 0101'. 0010'

Hexadecimal: 6 5 . 2

Ans: 65.2

- (iii) Evaluate the expression : $1 \ \&\& \ 0 \% 10 \ \geq \ 0 \ \&\& \ 30 \% 10 \ \leq \ 3$.
Show the steps of your expression evaluation.

[2]

Ans: 1

**Step 1: After arithmetic expression evaluation $1 \ \&\& \ 1 \ \&\& \ 1$
After logical expression evaluation: 1**

- (iv) Consider the definition `int test[2][5]`. How many `int` data will the array `test` contain? Explain your answer.

[1]

Ans: 10 int data, $2*5=10$

- (v) How many times will the following while loop execute? Explain your answer. [1]

```
char a='a';  
while(a > 'a' && a <= 'z')    a++;
```

Ans: 0 times. Because 'a' not greater than 'a'.

- (vii) What is the value of `x` after the execution of the following program segment? Explain your answer. [1]

```
x=-5; y=10;  
if(x>y)x=1;  
    else if(y<0) x=(x) * (-1);  
    else x=2*x;
```

Ans: $x=-10$, Both the if conditionals evaluate to false.

- (viii) What will be printed by the following code segment? Explain your answer. [1]

```
int i=0, j=45, x;  
x= i++ + ++j;  
printf("x=%d, i=%d j=%d \n",x,i,j);
```

Answer: $x=46, i=1, j=46$

Explanation: For `i++` the value contributed in the expression is 0 and for `++j`, its value in the expression is 46.

- (ix) What will be printed by the following code segment? Explain your answer. [1]

```
char c='r';  
switch(c)  
{  
    case 'b': printf("Blue \n");  
    case 'r': printf("Red \n");  
    case 'g': printf("Green \n");  
        break;  
    case 'y': printf("Yellow \n");  
    default:  
        printf("Other \n");  
}
```

Answer:

Red

Green

Explanation: As there is no 'break' after 'printf("Red \n");' in the switch statement.

(x) What will be printed by the following code segment? Explain your answer. [1]

```
int i=1;
if (i=0)
    printf("False \n");
else
    printf("True \n");
```

Answer:

True

Explanation: As the value of (i=0) returns 0 in the if clause.

Q.3) (a) Write a function named **cosine** which shall take a floating point variable x and evaluate the following series to the 5th decimal point of accuracy and returns the value as a double data type. Your program should compute each successive term based on the previously computed terms: [4 Marks]

$$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \dots$$

Answer:

```
double cosine(float x)
{
    int i;
    double term,sum;

    term=1;
    i=0;
    sum=1;

    while(term>.00001)
    {
        term= term*x*x/(double)((i+1)*(i+2));
        sum+=term;
        i=i+2;
    }

    return(sum);
}
```

Marking Guidelines: The implementation may vary and marks will be awarded if it is correct. If direct computation of factorial and exponentiation are carried out 2 marks will be deducted. The above function actually computes cos-hyperbolic function. If anyone answers by modifying the series for actual cosine computation, marks will be awarded.

- (b) Write a main program which reads **three** values for a triangle, namely lengths of its two sides (say, a and b), and the angle in radian formed by them (say, z). Using the above function and the formula given below, print the length of the other side (say c): [4 Marks]

$$c = \sqrt{a^2 + b^2 + 2 \cdot a \cdot b \cdot \cos(z)}$$

Answer:

```
#include <stdio.h>
#include <math.h>
main()
{
    double a,b,c,z;

    printf("Read a,b,z \n");
    scanf("%lf%lf%lf",&a,&b,&z);
    c=sqrt(a*a+b*b+2*a*b*cosine(z));
    printf("c=%lf \n",c);
}
```

Marking Guidelines: The implementation may vary and marks will be awarded if it is correct.

- (c) Write a C-function that receives a positive floating-point number as a parameter and returns the rounded value of the floating-point to two decimal places. For example, 248.765012 becomes 248.77. (*Hint: To round, you may convert the floating point number appropriately to an integer and then back to a floating point number.*) [4 Marks]

Answer:

```
float fround(float x)
{
    float y, fraction;
    int whole_number;

    y=100*x;
```

```

whole_number=(int) y;
fraction=y-(float) whole_number;

if (fraction>=0.5)
    whole_number=whole_number+1;

return((float)whole_number/100);
}

```

Marking Guidelines: The implementation may vary and marks will be awarded, if it is correct.

Please note that:

For all the above three parts, typical marks deductions are illustrated below:

- (i) Ignoring necessary type casting: $\frac{1}{2}$.
- (ii) Improper loop termination: 1.
- (iii) Any minor syntax error: $\frac{1}{2}$.
- (iv) Not implemented as function: 1.
- (v) A function without a return statement: 1.
- (vi) Printing or reading error: $\frac{1}{2}$.

Q. 4)a)Write a function **conv_cap** that takes a character string of arbitrary length as its input parameter and converts all the alphabets in the string to capital. Do not use any string or character library function. [3 Marks]

```

Ans: void conv_cap( char str[ ]){
        int i=0 ;
        while(str[i] !='\0'){
            if(str[i]>='a' && str[i]<'z') str[i]=str[i]-'a'+'A';
            i++;
        }
    }

```

Marking guideline: 0.5 mark for correct function prototype, 1 mark for correct loop, 1 mark for correct conversion, 0.5 mark for correct conditional.

b) Declare a global array named **studArr** consisting of 600 **student** structures, where each structure **student** has the following members: **Name:** string of 10 characters, **Address:** string of 20 characters, **Roll_number:** integer, **CGPA:** float. Write the code to read the student data for 600 students from terminal. Through call to the function **conv_cap** (part (a) of this question) convert the names of all students to uppercase and print those names. [4 Marks]

```

Ans: struct student{
        char Name[11];
        char Address[21];
        int Roll_number;
        float CGPA;
    } studArr[600]; /*studArray definition*/

```

```

int i;
for(i=0;i<600;i++){
    printf("Enter name address roll number and CGPA\n");
    scanf("%s%s%d%f", studArr[i].Name, studArr[i].Address,
        &studArr[i].Roll_number,&studArr[i].CGPA);
    printf("%s %s %d %f \n", studArr[i].Name,
        studArr[i].Address,studArr[i].Roll_number,studArr[i].CGPA);
}
for(i=0;i<600;i++){
    conv_cap(studArr[i].Name); /*Conversion function call*/
    printf("Converted name: %s \n", studArr[i].Name); /*output*/
}

```

Marking Guideline: Correct studArray definition 2 Marks, Correct input 0.5 Marks, Correct function call 1 Mark, Correct output 0.5 Marks.

c) There is a possibility that data corresponding to some students might have been entered multiple times. Write a function named **elimDuplicate** that takes no argument, and checks duplicate entries by matching the names in **studArr**. All those duplicate entries must be eliminated by moving forward the elements of **studArr** after a deleted entry. No additional array may be created for this purpose. After eliminating the duplicate student data, print out all the student data. [Hint: You may use any string library function, if you wish] [5 Marks]

```

void elimDuplicate(){
    int i,j,k,last=600;
    for(i=0; i<last-1; i++)
        for (j=last-1; j>i; j--)
            if (strcmp(studArr[i].Name,studArr[j].Name) ==0) {
                /*if match*/
                for(k=j; k<last; k++) studArr[k]=studArr[k+1]; /* shift rest one place forward*/
                last--; /* one record less */
            }
    for (i=0; i<last;i++)
        printf("Name=%s, Address=%s,Roll=%d,CGPA=%f\n",studArr[i].Name,
            studArr[i].Address, studArr[i].Roll_number,studArr[i].CGPA);
}

```

Marking Guideline: Correct loop formulations for matching 1.5 mark, correct string comparison 1 mark, correct shifting 1.5 Mark, array bound adjustment 0.5 mark, correct printing 0.5 Marks.

Q. 5) a) Write a C-function named **parking_fee** that takes the following two parameters – (i) Type of the vehicle, ('M' or 'm' for motorbike, 'C' or 'c' for car, and 'B' or 'b' for Bus), and (ii) Number of hours that a vehicle spent in the parking lot. The function should return the parking charge based on the following parking rates – Rs. 5, Rs. 10 and Rs. 50 per hour respectively for motorbike, car and bus. [3 Marks]

Answer:

```

int parking_fee(char ch, int parkHour){
    int parkFee;

    switch(ch)
    {

```

```

case 'M':
case 'm': parkFee = 5*parkHour; break;

case 'C':
case 'c': parkFee = 10*parkHour; break;

case 'B':
case 'b': parkFee = 50*parkHour; break;

default : printf("Unrecognized vehicle type . . .");
          parkFee = -1; break;
}
return parkFee;
}

```

Marking Guidelines: The implementation may vary; the above is a sample implementation.

If (correct function parameters, return, their types, and proper function body) Marks = 1;
 AND if (correctly compiled code with complete logic) Marks = 2;
 AND if (Correct Logic) Marks = 3.

b) Write a recursive C-function named **reverse** that reverses the elements of an integer array of maximum size 100, so that the last element becomes the first, the second from the last becomes the second, and so forth. Your function should take the integer array and the number of elements in the array as the parameters. The function should reverse the elements in place – that is, without using another array. [4 Marks]

Answer:

```

void reverseArray(int a[ ], int n)
{
    int temp;
    static int i =0;
    if (i>=n)
    {
        i=0; /* Initializing the static variable for next function call */
        return;
    }

    temp = a[i]; a[i]=a[n-1]; a[n-1]=temp; /* Swapping the corresponding elements */
    i++; /* Moving the left array index ahead for next recursive call */
    return reverseArray(a,n-1); /* Recursive call */
}

```

Marking Guidelines: The implementation may vary; the above is a sample implementation.

If (correct function parameters, return, their types and recursive function body) Marks = 1;
 AND if (correct base/terminating condition) Marks = 2;
 AND if (correctly compiled code with complete logic) Marks = 3;
 AND if (Correct Logic) Marks = 4.

c) Write a C-program to print a pyramid of digits (as shown below) for a given number of lines ($n < 10$) to be read from keyboard. For example, for $n=5$, the pyramid of 5 lines should be printed as: [5 Marks]

```

                1
              2 3 2
            3 4 5 4 3
          4 5 6 7 6 5 4
        5 6 7 8 9 8 7 6 5
    
```

Answer:

```
#include <stdio.h>
```

```
void printPattern(int n){
    int i,j;
```

```
    if (n<1) printf(" . . nothing to print\n");
```

```
    else{
```

```
        for (i=1; i<=n; i++)
```

```
        {
```

```
            for (j=i; j<n; j++) printf(" "); //get the start point to print
```

```
            for (j=0; j<i; j++) printf(" %2d",i+j); //print in increasing order
```

```
            for (j=j-2; j>=0; j--) printf(" %2d",i+j); //print in decreasing order
```

```
            printf("\n");
```

```
        }
```

```
    }
```

```
    return;
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    printf("Give number of lines (<10) \n");
```

```
    scanf("%d",&n);
```

```
    printPattern(n);
```

```
}
```

Marking Guidelines: The implementation may vary; the above is a sample implementation.

If (a compiled code having all functionality) Marks = 1;

AND if (correct code to print all the numbers correctly in each line) Marks = 3;

AND if (Correct Pattern) Marks = 5.

-----The End-----