User-Define Function

REVIEW QUESTION:

Question: State whether the following statements are true or false

(a) C function can return only one value under their function name.
Answer: False
(b) A function in C should have at least one argument.
Answer: True
(c) A function can be defined and placed before the main function.
Answer: False
(d) A function can be defined within the main function.
Answer: False.
(e) An user-defined function must be called at least once; otherwise
a warming message will be issue
Answer: True.
(f) Any name can be used as a function name.
Answer: False.
. (g) Only a void type function can have void as its argument
Answer: False.
(h) When variable values are passed to function, a copy of them are
created in the memory.
Answer: True.
(i) Program execution always begins in the main function irrespective of location in the program.
Answer: True.
(j) Global variable are visible in all blocks and function in the program
Answer: False.
(k) A function can call itself.
Answer: True
(l) A function without a return statement is illegal.
Answer: False.
(m) Global variable can not be declared as auto variables.
Answer: False.
(n) A function prototype must always be placed outside the calling function.
Answer: True
(o) The return type of a function int by default.
Answer: True.

(p) The variable names used in prototype should match those used in the function definition.
Answer: True.

(q) In parameter passing by pointers, the formal parameters must be prefixed with the symbol
* in their declarations.
Answer: True.

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

(r) In parameter passing by pointers, the actual parameters in the function call may be variables or constants.
Answer: False.

(s) In passing arrays to function, the function call must have the name of the array to be passed without brackets.
Answer: False.
(t) In passing strings to function, the actual parameter must be name of the strings post-fixed with size in brackets.
Answer: True.

Question: Fill in the blanks in the following statements.

(a) The parameters used in a function call are called …………….
Answer: actual parameter.
(b) A variable declared inside a function is called ……….. variable.
Answer: local
(c) By default…………is the return type of a function.
Answer: int
(d) In passing by pointers ,the variable of the formal parameters must be prefixed with …………….operator in their declaration.
Answer: indirection
(e) In prototype declaration specifying .parameter……….. is optional.
Answer: name
(f) …………… refers to the region where a variable is actually variable for use.
Answer: Prototype
(g) A function that calls itself is known as a…………. function.
Answer: recursive
(h) If a local variable has to retain its value between calls to the function, it must be declared as ………..
Answer: void
(i) A data ……….. aids the compiler to check the matching between the actual arguments and the formal ones .
Answer: types
(j) A variable declared inside a function by default assumes …………storage class.
Answer: without

Question: The main is a user-defined function. How does it differ from other user-defined function?

Answer:
The main is an example of user-defined function. printf and scanf belong to the category of library functions. We have also used other library functions such as sqrt, cos, strcat, etc. The main distinction between these two categories is that library functions are not required to be written by us whereas a user-defined function has to be developed by the user at the time of writing a program.

Question: Describe the two ways of passing parameters to function .When do you prefer to use each of them?

Answer:

The parameter list declares the variables that will receive the data sent by the calling program. They serve as input data to the function to carry out the specified task. Since they represent actual input values, they are often referred to as formal parameters. These parameters can also be used to send values to the calling programs. This aspect will be covered later when we discuss more about functions. The parameters are also known as arguments.
The parameter list contains declaration of variables separated by commas and surrounded by parentheses.
Examples;
float quadratic(int a, int b, int c) {….}

Question: What is prototyping? Why is it necessary?

Answer:

Prototypes enable the compiler to provide stronger type checking, somewhat like that provided by languages such as pascle. When you use prototypes, the compiler can find and report any questionable the conversions between the arguments used to call a function and the type of its parameters. The compiler will also catch differences between the number of arguments used to call a function and the number of parameters in the function. A function prototype consists of four parts.
(1)Function type
(2)Function name
(3)Parameter list
(4)Terminating semicolon.
The general form of a function prototype is function-type function-name (parameter list);

It is a good programming style to declare prototypes in the global declaration section before main. It adds flexibility, provides as excellent quick reference to the functions used in the program, and enhances documentation.

Question: Distinguish between the following:

(a)Actual and formal arguments

Answer:

If the actual parameters are more than the formal parameters the extra actual arguments will be discarded.
On the other hand, if the actual are less than the formals, the unmatched formal arguments will be initialized to some garbage.

(b) Global and local variables

Answer:

A global variable used in a function will retain its value for future use. A global variable is visible only form the point of its declaration to the end of the program. A local variable is a variable that is defined inside a function and used without having any role in the communication between functions.

(c) Automatic and static variables

Answer:

A variable declared inside a function without storage class specification default , an automatic variable. For instance, the storage class of the variable number in the example below is automatic.

```
main ( )
{
int number;
-------------
-------------
}
```

A static variable may be either an internal type or an external type. Depending on the place of declaration. Internal static variables are those which are declared inside a function .Therefore, internal static variables can be used to return values between function calls. For example, it can be used to count the number of calls made to a function.

(d) Scope and visibility variables

Answer:

The scope of variables determine s over what region of the program a variable is actually available for use (active) .on the other hand the visibility refers to the accessibility of a variable form the memory.

(e) & operator and * operator

Answer:

The operator & is called the address operator. On the other hand,

The operator * is known as indirection operator because it gives an indirect reference to a variable through its address.

Question: Explain what is likely to happen when the following situations are encountered in a program

.

(a)Actual arguments are less than the formal arguments in a function.

Answer: If the actual are less than the formals, the unmatched formal arguments will be initialized to some garbage.

(b) Data type of one of the actual arguments does not match with the type of the corresponding formal argument.

Answer: The formal and actual arguments must match exactly in type, order and number .their names, however, do not need to match.

(a)Data type of one of the argument in a prototype does not match with the type of the corresponding formal parameter in the header line.

Answer: Since the prototype is not available, c will assume that the return type is an integer and that the types of parameters match the formal definitions.
If these assumptions are wrong, the linker will fail and we will have to change the program. The moral is that we must always include prototype declarations preferably in global declaration section.

(a)The order of actual parameters in the function call is different from the order of formal parameters in a function where all the parameters are of the same type.

Answer:
The parameters used in prototypes and function definitions are called formal parameters and those used in function calls are called actual parameters. Actual parameters used in a calling statement may be simple constants , variables or expressions. The formal and actual parameters must match exactly in type , order and number . Their names, however do not need to match.

(a)The type of expression used in return statement does not match with the type of the function.

Question: Which of the following prototype declarations are invalid? Why?

(a) int (fun) void;
Answer: valid

(b)double fun (void)
Answer: double fun (void);

(c) float fun (x,y,n);
Answer: float fun (float x, float y, float n);

(d) void fun (void, void);
Answer: void fun (void);

(e)int fun (int a, b);
Answer: int fun (int a, int b);

(f)fun (int, float, char);
Answer: valid

(g)void fun (int a, int &b);
Answer: void fun (int a, int b);

Question: Which of the following header lines are invalid? Why?

(a) float average (float x, float y, float z);
Answer: valid
(b) double power (double a, int n-1)
Answer: double power (double a, double n-1);
(c) int product (int m, 10)
Answer: int product (int m, int 10);
(d) double minimum (double x; double y;)
Answer: double minimum (double x, double y);
(e) int mul (int x , y)
Answer: int mul (int x, int y);
(f) exchange (int *a, int *b)
Answer: exchange (int *a, int *b);
(g) void sum (int a, int b, int &c)
Answer: void sum (int a, int b ,int c);

Question: Find errors , if any, in the following function definitions:

```
( a)  void  abc ( int a, int b)
{
int  c;
.............
return  (c);
}
```

```
Answer: void abc (int a,int b)
{
int c;
-------------
}
```

```
(b)   int  abc ( int a, int b)
{
....................
........................
}
```

```
Answer: int abc (int a; int b)
{
int c;
-----------
return(c);
```

}

(c) int abc (int a, int b)

{
double c = a+b;
return (c);
}
Answer: int abc (int a, int b)
{
int c =a+b;
return (c);
}

(d) void abc (void)
{
....
....
return;
}

Answer: void abc (void)
{
int c;

}
(e) int abc (void)
{
....
....
return ;
}
Answer: Error in declaration.

Question: Find errors in the following function calls:
(a)void xyz();
Answer:
void xyz ( )
(b)xyx (void);
Answer:
void xyx (void)
(c)xyx (int x, int y);

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Answer:
int xyx (int x, int y)
(d)xyzz ();
Answer:
int xyzz( )
(e)xyz ()+xyz();
Answer:
int xyz ( ) + int xyz ( )
Question: A Function to divide two flowing point numbers is as follows:

```
divide (float  x,    float  y)
{

Return (x/y);
}
```

What will be the value of the following function calls

(a) divide (10,2)
Answer: 5.000000.
(b) divide (9,2)
Answer: 4.500000.
(c) divide (4.5,1.5)
Answer: 3.000000.
(d) divide (2.0,3.0)
Answer: .6777777.

Question: What will be the effect on the above function calls if we change the header line as follow:

(a) int divide (int x,int y)
(b) double divide(float x, float y)

Question: Determine the output of the following program?

```
int prod (int m, int n);
main ( )

{
int  x=10;
int  y=20;
int  p, q;
P=prod (x,y);
q=prod (p, prod (x,z));
printf  ("%d %d\n",p,q);
}
int prod (int a,int b)
{
return  (a*b);
}
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Answer: p=200
q =4000

Question: What will be the output of the following program?

```
Void test (int *a);
main( )
{
int  x=50;
test  ( &x);
printf (“%d\n”, x);
}
void  test  (  int  *a);
{
*a=*a +50;
}
```

Answer: 100.

Question: The function test is coded as follows:

```
int  test  ( int  number)
{
int m,n=0;
while  (number)
{
m= number % 10;
if ( m% 2 )
n=n +1;
number = number/10;
}
return ( n) ;
}
```

What will be the values of x and y when the following statement are executed
int x = test (135);
ans:x=3

int y= test (246);
Answer: y=o

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Question: Enumerate the rules that apply to a function call.

Answer:

A function call is a postfix expression. The operator (…) is at a very high level of precedence, therefore, when a function call is used as a part of an expression , it will be evaluated first , unless parentheses are used to change the order of precedence.

In a function call, the function name is the operand and the parameters set (..) which contains the actual parameters is the operator . The actual parameters must match the functions formal parameters in type, order and number. Multiple actual parameters must be separated by commas.

Question: Summarize the rules for passing parameters to functions by Pointers.

Answer:
The types of the actual and formal arguments must be same.
The actual arguments must be the addresses of variables that are local to the calling function.
The formal arguments in the function header must b\e prefixed by the indirection operator.
In the prototype the arguments must be prefixed by the symbol *.
To access the value of an actual argument in the called function, we must use the corresponding formal argument prefixed with the indirection operator *.

Question: What are the rules that govern the passing of arrays to functions.
Answer:
The function must be called by passing only the name of the array.
In the function definition, the formal parameter must be an array type, the size of the array does not need to be specified.
The function prototype must show that the argument is an array.

Question: State the problems we are likely to encounter when we pass global variables as parameters to functions

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Answer:
Since all functions in a program source file can access global Variables, they can be used for passing values between the functions. However, using global variables as parameters for passing values poses certain problems.

The values of global variables which are sent to the called function may be changed in advariently by the called functions.
Functions are supposed to be independent and isolated modules. This character is lost, if they use global variables.
It is not immediately apparent to the reader which values are being sent to the called function.
A function that uses global variables suffers from reusability.

Question:9.1 Write a function exchange to interchange the values of two variables ,say x and y. Illustrate the use of this function, in a calling function .Assume that x and y are defined as global variables.

Answer :

```
 #include<stdio.h>
void exchange_value(int I,int j);
void main  (void)
{
int x=2,y=3;
printf("%d %d", x,y);
exchange(x,y);
printf("%d %d",x,y);
}
void  exchange_value(int I,int j);
}
int  t; t=I                                                  ;
i=j; j=t;
printf("Exchange_value i=%d,j=%d",I,j);
}
```

Question: Write a function space(x) that can be used to provide a space of x positions between two output numbers .Demonstrate its application.

Answer:

```
#include<stdio.h>
void dinar(void);
void main()
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
{
 int x=2; ;
 int y=3;
 printf("%d",x);
 dinar();
 printf("%d",y);
}
void dinar(void)
{
 printf("  ");
```

***Question:9.3** Use recursive function calls to evaliuate*
F(x)=x-x3/3!+x5/5!-x7/7!+………

Answer:

```
#include<stdio.h>
#include<math.h>
double fact(int power)
      {
                    double f=1;
                    if(power==1)
                     return 1;
                    else
                     f=power*fact(power-1);
                    return f;
                    }

void main()
{
  int i=1;
  double x,term,deno,lob,sin,power=3;

  scanf("%lf",&x);
  term=x;
  sin=x;
  while(term>=0.0001)
  {
    lob=pow(x,power);
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
    deno=fact(power);
    term=lob/deno;
    power+=2;
    if(i%2==1)
      sin=sin-term;
    else
      sin=sin+term;
    i++;
  }
  printf("%lf",sin);
}
```

Question9.4: ***An n_order polynomial can be evaluated as follows:***
P=(….((a0x+a1)x+a2)x+a3)x+…+an)

Answer:

```
#include<stdio.h>
#include<conio.h>
typedef struct poly

{
int coeff;

int expo;

}p;

p p1[10],p2[10],p3[10];

void main()

{

int t1,t2,t3,k;
int read(p p1[10]);
int add(p p1[10],p p2[10],int t1,int t2,p p3[10]);
void print(p p2[10],int t2);
void printo(p pp[10],int t2);
t1=read(p1);
print(p1,t1);
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
t2=read(p2);
print(p2,t2);
t3=add(p1,p2,t1,t2,p3);
printo(p3,t3);
}
int read(p p[10])
{
int t1,i;
printf("\n Enter the total no of terms");
scanf("%d",&t1);
printf("\n Enter the coeff and expo in descending order");
for(i=0;i<t1;i++)
scanf("%d%d",&p[i].coeff,&p[i].expo);
return(t1);
}
int add(p p1[10],p p2[10],int t1,int t2,p p3[10])
{
int i,j,k;
int t3;
i=0,j=0,k=0;
while(i<t1 && j<t2)
{
if(p1[i].expo==p2[j].expo)
{
p3[k].coeff=p1[i].coeff+p2[j].coeff;
p3[k].expo=p1[i].expo;
i++;j++;k++;
}
else if(p1[1].expo>p2[j].expo)
{
p3[k].coeff=p1[i].coeff;
p3[k].expo=p1[i].expo;
i++;k++;
}
else
{
p3[k].coeff=p2[j].coeff;
p3[k].expo=p2[j].expo;
j++;k++;
}
}
while(i<t1)
{
p3[k].coeff=p1[i].coeff;
p3[k].expo=p1[i].expo;
i++;k++;
}
while(j<t2)
{
p3[k].coeff=p2[j].coeff;
p3[k].expo=p2[j].expo;
j++;k++;
}
t3=k;
return(t3);
}
void print(p pp[10],int term)
{
int k;
printf("\n\n Given Polynomial:");
for(k=0;k<term-1;k++)
printf("%dx^%d+",pp[k].coeff,pp[k].expo);
printf("%dx^%d",pp[k].coeff,pp[k].expo);
}
void printo(p pp[10],int term)
{
int k;
printf("\n\n The addition of polynomial:");
for(k=0;k<term-1;k++)
printf("%dx^%d+",pp[k].coeff,pp[k].expo);
printf("%dx^%d",pp[k].coeff,pp[k].expo);
}
```

***Question:9.5** Write afuntion to evaluate the polynomial ,using an array variable.Test it using a main program.*
The Fibpnacci numbers are defined recursively as follows:

F1=1
F2=1
Fn=Fn-1+Fn-2,n>2

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Write a function that will generate and print the first n Fibanacci numbers . Test the function for N=5,10,and15.

Answer::

```
#include<stdio.h>
int fib(int  i);
void main()
{  int F[10];F[0]=0,F[1]=1;
int fib(int  i)
{  for(i=2;i<10;++i)
F[i]=F[i-2]+F[i-1];
for(i=0;i<10;++i)
printf("%d\n",F[i]);
fib(i);
}return(fib);
}
```

Question:9.6 Write a function that will round a floating point number to an indicated decimal place .For example the number 17.457 would yeild the value 17.46 when it is rounded off to two decimal places .

Answer:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void liton(void);
void main()
{
 liton();
}
void liton(void)
{
char d[100];
 float f;
 int l,i;
  gets(d);
  l=strlen(d);
 for(i=0;i<l;i++)
 {
         if(d[i]=='.')
         {
            if(d[i+2]>=5)
            {
                  d[i+2]++;
          d[i+3]='\0';
                  break;
        }
           }
 }

f=atof(d);
printf("%.2f",f);
 }
```

Question: 9.6 Write a function prime that returns l if its argument is a prime number and returns zero otherwise .

Answer:

```
#include<stdio.h>
void  isprime(int  n);
void main()
{ int  I,j,k;
for(i=3;i<100;i+=2)
{  If(isprime(i==1))
{ k=0;j=I;
while(j>0)
{ k+=j%10;   j=j/10;
}
if(isprime(k==1))
printf("%d",i);
}
}
} void  isprime(int  n)
{ int  m; for(m=2;m<=(n/2);m++)
{ if(n%m==0)
{ break;
}
```

Question: 9.8
Write a function that will scan a character string passed as an argument and convert all lowercase characters into their uppercase equivalents.

Answer:

```
#include<stdio.h>
#include<ctype.h>
char lower_to_upper(char c1);
{char c2;
C2=(c1>=’a&&c1<=’z’)?(‘A’+c1-‘a’):c2;
return(c2);
main()
{ char lower,upper;
printf(“enter a lowercase letter”);
scanf(“%c”,&lower);
upper=lower_to_upper(lower);
printf(“uppercase  equivalent  is %c\n ”,upper);
}
```

Question:9.9 Develop a top doun modular program to implement a calculator .The program should request the user to input two numbers and display one of the following as per the desire of the user:
(a) sum of the numbers
(b) Difference of the numbers
(c) Product of the numbers
.. (d) Division of the numbers
Provide separate functions for performing various tasks such as reading ,cakculating and displaying. Calculating module should call second level moduls to perform the individual mathematical operations. The main function should have only function calls.

Answer: (a)

```
#include<stdio.h>
void sum (int I,int j) ;
void main()
{
sum(5,6);
}
void sum(int I, int j);
{
int total;
Total=i+j;
printf("%d",total);
}
```

(b)

```
#include<stdio.h>
void diff (int I,int j) ;
void main()
{
diff(5,6);
}
void diff(int I, int j);
{
int total;
Total=i-j;
printf("%d",total);
}
```

(c)

```
#include<stdio.h>
void prod (int I,int j) ;
void main()
{
product(5,6);
}
void product(int I, int j);
{
int total;
Total=i-j;
printf("%d",total);
}
```

(d)

```
#include<stdio.h>
void div (int I,int j) ;
void main()
{
div(5,6);
}
void div(int I, int j);
{
int total;
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
Total=i-j;
printf("%d",total);
}
```

Question: 9.10

Develop a modular interactive program using functions that resds the values of three sides of triangle and displays either its area or its perimeter as per the request of the user. Given the three sides,a,b,c.

Perimeter=a+b+c

Area=sqrt(s-a)(s-b)(s-c) Wher s=(a+b+c)/2

sides,a,b,c.

Perimeter=a+b+c

Area=sqrt(s-a)(s-b)(s-c) Wher s=(a+b+c)/2

Answer:

```
#include<stdio.h>
void area(int a,int b,int c);
void  perimeter(int a,int b,int c);
void  main()
{
   printf("input  a,b,c");
scanf("%d  %d  %d",&a,&b,&c);
}
  void  perimeter(int a,int b,int c)
{
  int  s;    s=(a+b+c)/2;
printf("%d",s);
}
void  area(int  a,int b,int c,int s)
{
   float  A;  A=sqrt((s-a)*(s-b)*(s-c));
  printf("%f",A);
}
```

Question: 9.11

Write a function that can be called to find the largest element of an m by n matrix.

Answer:

```
 #include<stdio.h>
void max_ele(int r,int c,int dinar[][11]);
void main()
{
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
   int sazon[11][11]={0},k;
   int r,c,i,j;
   printf("Enter nomber of rows and columns(Maximun number can be 11):");
   scanf("%d%d",&r,&c);
   printf("Enter the elements of the matrix in rowwise:\n");
   for(i=0;i<r;i++)
     for(j=0;j<c;j++)
     {
          scanf("%d",&k);
          sazon[i][j]=k;
     }
  max_ele(r,c,sazon);
}
void max_ele(int r,int c,int sazon[][11])
{
   int max,i,j;
   max=sazon[0][0];
   for(i=0;i<r;i++)
    for(j=0;j<c;j++)
    {
       if(max<sazon[i][j])
       {
           max=sazon[i][j];
       }
    }

   printf("\nThe maximum elements is:%d",max);
}
```

Question:9.12 Write a function that can be called to compute the product of two matrixes of size m by n and n by m.The main function provides the values for m and n and two matrices.
main function provides the values for m and n and two matrices.

Answer:

```
 #include<stdio.h>
void  prod(int  a[4][4] ,int  b[4][4]);
void main()

{   int  r,q,m,n,I,j,y;
printf("input  row  and  column  of  A  matrix");  scanf("%d  %d", &m, &n);
printf("Input  row  and  column  of  B matrix");   scanf("%d  %d",&q ,&r);
if(n==q)
printf("A and B  can  multiplied");  printf("resultant  matrix  is  %d *%d",m,r);
printf("input  A  matrix");    mat(a,m,n);
printf("input  B  matrix");    mat(b,q,r);
void  prod(int  a[4][4] ,int  b[4][4])
{
 for(i=0; i<m; ++i)    for(j=0; j<q; ++j)
C[i][j]=0;   for(y=0;y<n;++y)
C[i][j]=c[i][j+a[i][y] *b[y][j];
}
  printf("resulatant  of  A  and  B  matrix");   matrix(c,m,r);
}
  else
{
      printf("A      col      be      equal      to      row      0f      B      matrix"
);  printf("matrixes  can  not  be  multiplied");
}
   mat(a,m,n);      int  a[4][4],m,n;
}
   int  I,j;   for(i=0; i<m; i++)    for(j=0; j<n; j++)
 scanf("%d",&a[i][j]);
}
  mat(a,m,n);
{
      int  a[4][4],m,n;
}
   int  I,j;   for(i=0; i<m; i++)    for(j=0; j<n; j++)
 printf("%d",a[i][j]);
    }
}
```

Question: Design and code an interactive modular programming that will use function to a matrix of m by n size. compute column average and rows aver ages , and then print the entire matrix with averages shown in respective rows and columns.

Answer:

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Question: 9.14

Develop a top down modular program that will perform the following tasks
(a)Read two integer arrays with unsorted element (b)Sort them in accending order
(c)Merge the sorted arrays (d)Print the sorted list
Use function for carrying out each of the above tasks. The main function should have only function calls.

Answer:

```
#include<stdio.h>
int  sort (int  a[8], int  b[8]);
void main()
{  int  I,j;  printf("array  before  sort");
printf("enter  dimension"');   scanf("%d",&j);
printf("enter  %d  values  for  a , j");
for(i=0;  i<j; i++)   scanf(%d",&a[i]);
 printf("enter  %d  values  for  b , j");
for(i=0;  i<j; i++)   scanf(%d",&b[i]);
printf("element  array  of  a");
for(i=0;  i<j; i++)   scanf(%d\t",&a[i]);
printf("element  array  of  b");
for(i=0;  i<j; i++)   scanf(%d\t",&b[i]);
for(i=0;  i<j; i++)
{    a[i]=a[i]+b[i];   b[i]=a[i]-b[i];
  a[i]=a[i]+b[i];      sort(a[8],b[8]);
}    printf("after  sorting ");
int  sort (int  a[8], int  b[8])
{    printf("element  array  of  a");
     for(i=0;  i<j; i++)   scanf(%d\t",&a[i]);
printf("\n\n\n\n\n\n");
Printf("element  array  of  b");
for(i=0;  i<j; i++)   scanf(%d\t",&b[i]);
return(sorted  list);
}
```

Question: 9.15

Develop your owx functions for performing following operations on strings:
(a) copying one string to another
(b) comparing two strings
(c) adding a string to the end of another string
Write a driver program to test your functions.

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

functions.

Answer:

(A)

```
#include<stdio.h>
#include<string.h>
char  *strcpy(char *str1,char *str2);
void main()
{ strcpy(str1,str2);
return(str1,str2);
} char *strcpy(char*str1,char*str2)
{ char str1[10];char str2[10]=”murad”;
puts(str1);  puts(str2);
}
```

(B)

```
#include<stdio.h>
#include<string.h>
char  *strcmp(char *str1,char *str2);
void main()
{ strcmp(str1,str2);
return(str1,str2);
} char *strcmp(char*str1,char*str2)
{ char str1[10];char str2[10]=”murad”;
gets(str1);   gets(str2);
}
```

(C)

```
include<stdio.h>
#include<string.h>
char  *strcat(char *str1,char *str2);
void main()
{ strcat(str1,str2);
return(str1,str2);
} char *strcat(char*str1,char*str2)
{ char str1[10];char str2[10]=”murad”;
gets(str1);  gets(str2);
puts(b);
}
```

Question:9.16 Write a program that invokes a function called find() to perform the following tasker.

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

(a) Receives a character array and a single character.
(b) Returns 1 if the specified character is found in the array ,0 otherwise.

Answer:

```
#include<stdio.h>
void  find(int  num[8]);
void main()
{   int a[8],i;
for(i=0;  i<=8;  ++i)
Printf("%d",num[j];
  if(a[i])=i;
find(a[i]);
getch();
}
void  find(int num[8])
{   int j;
if(a[i])=i)
return(1);
else
return(0);
}
printf("%d",num[j];
}
```

Question: 9.17

Design a function locate () that takes two character arrays s1 and s2 and one integer value m as parameters and inserts the strings s2 into s1 immediately after the index m.Write a program to test the function using a real-life situation.(Hints :s2 may be a missing word in s1 that represents a line of text).

Answer:

```
 #include<stdio.h>
#include<string.h>
void lokate(void);

void main()
{
        lokate();
}
void lokate(void)
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
{

        char s1[40],s2[10],s3[30];
        int m,l1,l2,i,j,k;
        printf("Input string s1:");
        gets(s1);
        printf("Input string s2:");
        gets(s2);
        l1=strlen(s1);
        l2=strlen(s2);
        printf("Input position where s2 is missed from s1:");
        scanf("%d",&m);

        i=0;j=m;

        for(m=j-1;m<l1;m++)
        {
                s3[i++]=s1[m];
        }
        //for s3

        for(k=0;k<l2;k++)
        {
         s1[j++]=s2[k];
        }
        //j--;
         for(k=0;k<i;k++)
         {
                s1[j++]=s3[k];
         }
         s1[j]='\0';
         puts(s1);


}
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

Question:9.18

Write a function that takes an integer parameter m representing the month number of the year and returns the corresponding name of the month .For instants ,if m=3, the month is March.Test your program.

Answer:

```
 #include<stdio.h>
void month(int n,char a[]);
void main()
{
        int k;
        char dinar[15]="";
        printf("Input month number:");
        scanf("%d",&k);
        month(k,dinar);
        printf("%s",dinar);
}
void month(int n,char a[])
{
         switch(n)
         {
                 case 1:strcpy(a,"January");
                                 break;
                 case 2:strcpy(a,"Fabruary");
                                 break;
                 case 3:strcpy(a,"March");
                                 break;
                 case 4:strcpy(a,"April");
                                 break;
                 case 5:strcpy(a,"May");
                                 break;
                 case 6:strcpy(a,"June");
                                 break;
                 case 7:strcpy(a,"July");
                                 break;
                 case 8:strcpy(a,"August");
                                 break;
                 case 9:strcpy(a,"September");
                                 break;
                 case 10:strcpy(a,"October");
```

```
                                        break;
                        case 11:strcpy(a,"November");
                                        break;
                        case 12:strcpy(a,"December");
                                        break;
            }
}
```

Question: 9.19
In preparing the calander for a year we need to know whether the particular year is leap year or not .Design a function leap () that receives the year as a parameter and returns an appropriate messege.
What modifications are required if we want to use the function in preparing the actual calender?
Answer:

```
#include<stdio.h>
int  leap(int  y);
void  main()
{  printf("input a  year");     scanf("%d",&y);
If(((y%4)==0)&&((y%100)!=0)  || ((y%400)==0))
Printf(" This is a leap year");
else
 printf(" This is not a leap year");

leap (y);
}     int leap(int  y);
{  return(y);
}
```

Question:9.20
Write a function that receives a floating point value x and returns it as a value rounded to two nearest decimal place.For example ,the value 123.4567 will be rounded to 123.46(Hint:Seek help of one of the math functions available in math library).
Answer:

```
#include<stdio.h>
#include<math.h>
float dec_num(float a);
void main()
{    printf("  enter  real  number  to get  nearest  in teger number ");
```

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering

```
   scanf("%f",&a);
float dec_num(float a)
{  if(a>=0)
P=a+0.5;
else
p=a-0.5;
}
return(int(p));
}
```

Assignments:

1. Write the difference between User defined function and Built in function?
2. Write down the category of functions.
3. Write a program for Matrices Manipulation using functions.
4. Write a program to find the factorial value of a given number by using Recursive function.

Drs. UtpalKanti Das
Faculty & Coordinator
Department of Computer Science and Engineering