COMPUTER SCIENCE

Paper – 2

(Planning Session : One hours)
(Examination Session : Two hours)

Instructions

As it is a practical examination the candidate is expected to do the following:

(i) Write an algorithm for the selected problem

(ii) Write a program in C++/Java. Document your program by using mnemonic names and comments

(iii) Test run the program on the computer using the given test data and get a print out (hard copy) in the format specified in the problem along with the program listing.

Solve any one of the following problem.

Question 1.

Write a program which takes a string (maximum 80 characters terminated by a full stop. The words in this string are assumed to be separated by one or more blanks.

Arrange the words of the input string in descending order of their lengths. Same length words should be sorted alphabetically. Each word must start with an uppercase letter and the sentence should be terminated by a full stop.

Test your program for the following data and some random data.

SAMPLE DATA:

INPUT:

“This is human resource department.”

OUTPUT:

Department Resource Human This Is.

INPUT:

“To handle yourself use your head and to handle others use your heart.”

OUTPUT:

Yourself Handle Handle Others Heart Head Your Your And Use Use To To.

Question 2.

A wondrous square is an n by n grid which fulfils the following conditions:

(i) It contains integers from 1 to n^2, where each integer appears only once.

(ii) The sum of integers in any row or column must add up to $0.5 * n * (n^2 + 1)$.

For example the following grid is a wondrous square where the sum of each row or column is 65 when n = 5 :

17	24	2	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Write a program to read n (2 <= n <= 10) and the values stored in these n by n cells and output if the grid represents a wondrous square or not.

Also output all the prime numbers in the grid along with their row index and column index as shown in the output. A natural numbers is said to be prime if it has exactly two divisors. E.g. 2, 3, 5, 7, 11,…….

The first element of the given grid i.e. 17 is stored at row index 0 and column index 0 and the next element in the row i.e. 24 is stored at row index 0 and column index 1.

Test your program for the following data and some random data.

SAMPLE DATA:

INPUT:

```
N = 4
            16      15      1       2
            6       4       10      14
            9       8       12      5
            3       7       11      13
```

OUTPUT:

```
Yes it represents a wondrous square.
Prime       Row index       column index
   2            0               3
   3            3               0
   5            2               3
   7            3               1
   11           3               2
   13           3               3
```

INPUT:

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

OUTPUT:

```
Not a wondrous square
Prime       Row index       column index
   2            0               1
   3            1               0
   5            1               2
   7            1               1
```

Question 3.

We would like to generate all possible anagrams of a word. For example if the given word is ‘TOP’, there will be 6 possible anagrams:

```
TOP
TPO
OPT
OTP
PTO
POT
```

An anagram must be printed only once. You may output the anagrams in any order. Also output the total number of anagrams. You assume that the number of letter, n, in the word will be 7 at most, i.e. n<=7

Test your program for the given data and some random data.

SAMPLE DATA:

INPUT:

```
TO
```

OUTPUT:

```
TO
OT
Total number of anagrams = 2
```

```
INPUT:
LEAN
OUTPUT:
LEAN
LENA
LAEN
LANE
LNEA
LNAE
EALN
EANL
ELAN
ELNA
ENLA
ENAL
ALNE
ALEN
ANLE
ANEL
AENL
AELN
NLEA
NLAE
NELA
NEAL
NALE
NAEL
Total number of anagrams = 24
```

PROGRAM 1:

```
/* QUESTION 1 */
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include<ctype.h>
void main()
{ clrscr();
  int l,k=0,k1=0,l1,l2,j;
  char pp,str[80],d[30][90],temp[90];
  cout<<"Enter the string(<80) :";
  gets(str);
  l=strlen(str);
  for(int i=0;i<l;i++)
  { if(!isspace(str[i]))
      { d[k][k1]=str[i];
         k1++;
         }
  else
  {  d[k][k1]='\0';
     k++;
     k1=0;
  }
 }
 for(i=0;i<k;i++)
 { for(j=0;j<k-1;j++)
  { l1=strlen(d[j]);
   l2=strlen(d[j+1]);
   if(l1<l2)
   { strcpy(temp,d[j]);
   strcpy(d[j],d[j+1]);
   strcpy(d[j+1],temp);
   }
  }
 }
for(i=0;i<k;i++)     // to check for the words of equal lengths
 { for(j=0;j<k-1;j++)
  { l1=strlen(d[j]);
   l2=strlen(d[j+1]);
   if(l1==l2)
   { int q=d[j][0];
   int q1=d[j+1][0];
   if(q>q1)
   { strcpy(temp,d[j]);
    strcpy(d[j],d[j+1]);
    strcpy(d[j+1],temp);
   }
   }
  }
 }
 for(i=0;i<k;i++)   // for displaying
 { int p=strlen(d[i]);
  char u=d[i][0];
  if(islower(u))
  { pp=toupper(u);
   cout<<pp;
  }
  else
  cout<<d[i][0];
  for(j=1;j<p;j++)
  cout<<d[i][j];
  cout<<" ";
 }
 cout<<".";
 getch();
}
```

PROGRAM 2:

```
#include<iostream.h>
#include<conio.h>
void main()
{
 clrscr();
 int n,a[10][10],b[100],i,j,x1=1,y,sum1,sum2,x2=1,k,l,c;
 cin>>n;                        //Enter value of n
 for(i=0;i<n;++i)               //Enter array
  for(j=0;j<n;++j)              cin>>a[i][j];
 for(i=1;i<=n*n;++i)             b[i]=0;              //Assign single dimensional array to 0
for(i=0;i<n;++i)                            //Check validity of first condition
  for(j=0;j<n;++j)
  {               if(b[a[i][j]]==0)        b[a[i][j]]=1;
                 else x1=0;
  }
 if(x1==0) cout<<"NOT A WONDROUS SQUARE"<<endl;
 else
 {    y=0.5*n*(n*n+1);                 //Check validity of second condition
      for(i=0;i<n;++i)
     {
        sum1=0;
        sum2=0;
       for(j=0;j<n;++j)
   {     sum1=sum1+a[i][j];          //Stores sum of rows
         sum2=sum2+a[j][i];          //Stores sum of columns
   }
   if(sum1!=y || sum2!=y)  x2=0;
  }
  if(x2==0) cout<<"NOT A WONDROUS SQUARE"<<endl;
 }
 if(x1= =1 && x2= =1)
                         cout<<"YES IT REPRESENTS A WONDROUS SQUARE"<<endl;
 //Display primes.
 cout<<"PRIME \t ROW INDEX \t COLUMN INDEX"<<endl;
 for(i=2;i<=n*n;++i)
 {    c=0;
  for(k=1;k<=i;++k)
      if(i%k==0)    c++;
      if(c= =2)           //Number is Prime only if c=2
      {  for(j=0;j<n;++j)
          for(l=0;l<n;++l)
           if(a[j][l]==i)
            {   cout<<i<<"\t \t"<<j<<"\t \t"<<l<<endl;          }
   }       //End of if
 }       //End of for
 getch();
}       //End of program
```

PROGRAM 3:

```
//pemutation
#include <iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
perm(char a[],int k,int n)
 {int i,t;
 if(k==n) {
        for(i=0;i<=n;i++)
        cout<<a[i];
        cout<<endl;
        }
     else
      for(i=k;i<=n;i++)
       {
        t=a[k];
        a[k]=a[i];
        a[i]=t;
        perm(a,k+1,n);
        t=a[k];
        a[k]=a[i];
        a[i]=t;
       }
  }
void main()
{
 clrscr();
 int k,n,i;
 char a[10];
 cout<<"\n INPUT WORD or NUMBERS FOR GENERATING PERMUTATION ";
 cout<<"\n size is maximum 10 digits or characters";
 gets(a);
 k=0;
 n=strlen(a)-1;
 perm(a,k,n);
 getch();
 }
```