Programming for Problem Solving (PPS)

D.K. Sunil Kumar Reddy
T20CSB054

All Lab cycles 1 to 12 completed

Lab Cycles:-

lab1:- For the given basic salary, Compute DA, HRA & PF and write a program to find out net salary of an employee by deducting PF & IT.

```
#include <stdio.h>
void main()
{int b;
 float d, h, g, p, i, n;
 printf("enter basic salary of employee \n");
 scanf("%d", &b);
 d=0.25*b;
 h=0.15*b;
 g=b+d+h;
 p=0.10*g;
 i=0.10*g;
 n=(b+d+h)-(p+i);
 printf("The net salary of employee is Basic salary
      + DA + HRA - PF - IT = %d + %f + %f
      %f - %f = %f", b, d, h, p, i, n);
}
```

output:

enter basic salary of employee

25000

The net salary of employee is Basic salary + DA + HRA - PF - IT = 25000 + 6250 + 3750 - 3500 - 3500 = 28000

b) Write a program to convert the given minutes into days, hours, and remaining minutes.

Eg: M = 2000 Answer is 2000 Minutes contain 1 day 9 hours 20 minutes

```
#include <stdio.h>
void main()
{
    int min, hours, days;
    Printf("enter the minutes\n");
    Scanf("%d", &min);
    hours = min/60;
    min = min%60;
    days = hours/24;
    hours = hours%24;
    Printf("%d days, %d hours, %d minutes",
                days, hours, min);
}
```

Output:
enter the minutes
3000
2 days, 2 hours, 0 minutes

Output2:
enter the minutes
8000
5 days, 13 hours, 20 minutes

Lab 2:- Problems involving if-then-else & switch

a). The marks obtained by a student in 'n' different subjects is input. Write a c program to calculate average marks and display grade.

average	grade
90-100	O
80-89	(E),
70-79	(A),
60-69	(B)
50-59	(C)
<50	(F).

```
#include <stdio.h>
void main()
{ int i, n, sum=0, a[10];
  float avg;
  Printf ("enter number of subjects \n");
  Scanf ("%d", &n);
  Printf ("enter marks in each subject \n");
  for(i=0; i<n; i++)
  {Scanf ("%d", &a[i]);
  sum= sum+ a[i];
  }
  avg = sum/n;
  Printf (" The average marks of student is %of \n", avg
  if (avg >=90 && avg<=100)
  Printf ("The grade obtained is 'O' \n");
  else if (avg >=80 && avg<=89)
  Printf ("The grade obtained is 'E' \n");
  else if (avg >=70 && avg<=79)
  Printf ("The grade obtained is 'A' \n");
  else if (avg >=60 && avg<=69)
```

```
Printf ("The grade obtained is 'B' \n");
else if (avg >= 50 && avg <= 59)
Printf ("The grade obtained is 'C' \n");
else
Printf ("The grade obtained is 'F' \n");
}
```

Output 1:

```
enter number of subjects
5
enter marks in each subject
90
80
85
86
76
The average marks of student is 83.400000
The grade obtained is 'E'
```

Output 2:

```
enter number of subjects
6
enter marks in each subject
90
80
72
65
76
85
The average marks of student is 78.000000
The grade obtained is 'A'
```

b) Write a C program to find the largest and smallest values among the given 4 values:

```
#include <stdio.h>
Void main( )
{
  int a,b,c,d [illegible];
  Printf ("enter the four numbers \n");
  Scanf ("%d %d %d %d %d", &a, &b, &c, &d);
  if (a>b && a>c && a>d)
  Printf ("Biggest number is %d\n", a);
  else if (b>c && b>d)
  Printf ("Biggest number is %d\n", b);
  else if (c>d)
  Printf ("Biggest number is %d\n", c);
  else
  Printf ("Biggest number is %d\n", d);
  if (a<b && a<c && a<d)
  Printf ("smallest number is %d\n", a);
  else if (b<c && b<d)
  Printf ("smallest number is %d\n", b);
  else if (c<d)
  Printf ("smallest number is %d\n", c);
  else
  Printf ("smallest number is %d\n", d);
}
```

enter the four numbers	enter the four numbers
15	20
17	30
12	100
32	25
Biggest number is 32	Biggest number is 100
smallest number is 12	Smallest number is 20

c) Write a C program to print the month name when month number is given and printf the week name when week number is given.

```
# include <stdio.h>
Void main( )
{
int m, w;
Printf("enter month number \n"); scanf("%d", &m);
switch(m)
{
case 1 : Printf("January \n");
break;
Case 2 : Printf("Febraury \n");
break;
Case 3 : Printf("March \n");
break;
Case 4 : Printf("April \n");
break;
Case 5 : Printf("May \n");
break;
Case 6 : Printf("June \n");
break;
Case 7 : Printf("July \n");
break;
Case 8 : Printf("August \n");
break;
Case 9 : Printf("September \n");
break;
Case 10 : Printf("October \n");
break;
```

```
        case 11 : printf ("November\n");
        break;
        case 12 : printf ("December\n");
        break;
default : printf ("error\n");
break;
printf ("enter week number\n");
scanf ("%d", &w);
switch (w)
{
    case 1 : printf ("Sunday\n");
    break;
    case 2 : printf ("Monday\n");
    break;
    case 3 : printf ("Tuesday\n");
    break;
    case 4 : printf ("Wednesday\n");
    break;
    case 5 : printf ("Thursday\n");
    break;
    case 6 : printf ("Friday\n");
    break;
    case 7 : printf ("Saturday\n");
    break;
default : printf ("error\n");
break;
}
```

Output1:

```
enter month number
12
December
enter week number
7
Saturday
```

Output2:

```
enter month number
6
June
enter week number
2
Monday
```

8) Write a C program to check whether the given character is numeric or non-numeric

~~#include <stdio.h>~~
~~void main()~~

```
#include <stdio.h>
void main()
{
  char ch;
  printf("enter character\n");
  scanf("%c", &ch);
  if(ch>='a' && ch<='z' || ch>='A' && ch<='Z')
/*(ch>=97 && ch<=122 || ch>=65 && ch<=90) */
  printf("non numeric\n");
  else
  printf("numeric\n");
}
```

output:
enter character
b
non numeric

output 2:
enter character
98
numeric

Lab 3: Iterative Problems:

a) Write a C program to read in two numbers x and n, then compute sum of geometric progression

```
#include <stdio.h>
#include <math.h>
void main()
{
    int x, n, sum=0, i;
    Printf("enter values of x and n\n");
    Scanf("%d %d", &x, &n);
    if(n<0)
    Printf("n should be positive\n");
    else
    {
        for(i=0; i<=n; i++)
        sum = sum + pow(x, i);
        Printf("x=%d\n n=%d\n sum=%d\n", x, n, sum);
    }
}
```

---

output1:
enter values of x and n
5
6
x=5
n=6
Sum=19531

---

output2:
enter values of x and n
5
-6
n should be positive

b) Write a c program to print Floyd triangle for the user given number of rows

```
# include <stdio.h>
Void main()
{
  int i, j, n, K=0;
  Printf ("enter number of rows \n");
  scanf ("%d", &n);
  for (i=1; i<=n; i++)
  {
    for (j=1; j<=i; j++)
    {
      Printf ("%3d", ++K);
    }
    Printf ("\n");
  }
}
```

Output:

```
enter number of rows
5
  1
  2  3
  4  5  6
  7  8  9  10
 11 12 13  14  15
```

c) Write a C program to check whether the given number is a palindrome or not.

```
#include <stdio.h>
void main()
{
  int n, m, rev=0, rem;
  Printf ("enter n value \n");
  Scanf ("%d", &n);
  m=n;
  While (m!=0)
  {
    rem = m%10;
    rev = (rev*10) + rem;
    m= m/10;
  }
  If (n==rev)
  Printf ("The given no is palindrome\n");
  else
  Printf ("The given no is not palindrome\n");
}
```

output 1:

```
enter n value
121
The given no. is palindrome
```

output 2:

```
enter n value
658
The given no. is not palindrome
```

d) Write a c program to find all prime numbers, strong numbers and armstrong numbers between the given two numbers.

```
#include <stdio.h>
#include <math.h>
void main()
{
int a,b,i,c,n,m,arm,p,j,f,s,r1,r2,u,k;
Printf("enter two numbers \n");
scanf("%od %od", &a, &b);
scanf("%od", &k);
Printf("enter 1 for prime numbers, 2 for armstrong, 3 for strong numbers \n");
Scatch(k)
{
case 1: for(u=a; u<=b; u++)
        {
          c=0;
          for(i=0; i<=u; i++)
          {
            if(u%i==0)
              c++;
          }
          if(c==2)
          Printf("%od,", u);
        }
        Printf("are prime numbers \n\n");
break;
```

```
case 2 : for (u=a; u<=b; u++)
         {
           c=0, arm=0;
           for (n=u; n!=0; n=n/10)
           c++;
           for (m=u; m!=0; m=m/10)
           {
             r1 = m%10;
             arm= arm + pow(r1, c);
           }
           if (arm == u)
           printf ("%d", arm);
         }
         Printf (" are armstrong numbers \n");
break;

case 3 : for (u=a; u<=b; u++)
         {  s=0;
            for (p=u; p!=0; p=p/10)
            {
              r2= p%10;
               f=1;
              for (j=r2; j!=0; j--)
              f= f * j;
              s= s+f;
            }
            if (s==u)
            printf ("%d", u);
```

```
}
Printf("a are strong numbers\n");
break;
default: Printf("error\n");
break;
}
}
```

Output1

enter two numbers
10
20
enter 1 for prime numbers, 2 for armstrong, 3 for strong numbers
1
11, 13, 17, 19, are prime numbers

Output 2:
enter two numbers
140
160
enter 1 for prime numbers, 2 for armstrong, 3 for strong numbers
2
153, are armstrong numbers

Output3:
enter two numbers
1
150
enter 1 for prime numbers, 2 for armstrong, 3 for strong numbers
3
1, 2, 145, are strong numbers.

e, Convert rupees into text:

```
# include <stdio.h>
void main()
{
   int n, m, r[5], i=0;
   Printf("enter rupees \n");
   Scanf("%d", &n);
   while(n!=0)
   {
      i++;
      r[i]=n%10;
      n = n/10;
   } if(i==0) Printf("zero");
else if(i>=5)
   Printf("enter number upto 4 digits only \n");
else
   {
      switch(r[4])
      {
         case 1: Printf("One");
         break;
         case 2: Printf("Two");
         break;
         case 3: Printf("Three");
         break;
         case 4: Printf("Four");
         break;
```

output:
enter rupees
150
One Hundred and Fifty Rupees only

output:
enter rupees
4025
Four Thousand Twenty Five Rupees Only

```
case 5 : Printf ("Five");
break;
case 6 : Printf ("Six");
break;
case 7 : Printf ("Seven");
break;
case 8 : Printf ("Eight");
break;
case 9 : Printf ("Nine");
break;
default : Printf ("Error");
case 0 : Printf (" ");
break;
}
if(i==4
Printf (" Thousand");
switch(r[3])
{
case 1 : Printf (" One");
break;
case 2 : Printf (" Two");
break;
case 3 : Printf (" Three");
break;
case 4 : Printf (" Four");
break;
case 5 : Printf (" Five");
break;
```

```
Case 6 : Printf (" Six");
break;
Case 7 : Printf (" Seven");
break;
Case 8 : Printf (" Eight");
break;
Case 9 : Printf (" Nine");
break;
~~Case 0: Printf("");~~
~~break;~~
}
if (i>=3 && r[3]!=0)
Printf (" Hundred and");
Switch(r[2])
{
  case 1 : switch (r[1])
           {
             case 0 : Printf (" Ten");
             break;
             case 1 : Printf (" Eleven");
             break;
             case 2 : Printf (" Twelve");
             break;
             Case 3 : Printf (" Thirteen");
             break;
             Case 4 : Printf (" Fourteen");
             break;
             Case 5 : Printf (" Fifteen");
```

```
                    break;
                    case 6 : Printf (" Sixteen");
                    break;
                    case 7 : Printf (" Seventeen");
                    break;
                    case 8 : Printf (" Eighteen");
                    break;
                    case 9 : Printf (" Ninteen");
                    break;
                }
                break;
case 2 : Printf (" Twenty");
  break;
case 3 : Printf (" Thirty");
 break;
case 4 : Printf (" Forty");
break;
case 5 : Printf (" Fifty");
break;
case 6 : Printf (" Sixty");
break;
case 7 : Printf (" Seventy");
break;
case 8 : Printf (" Eighty");
break;
case 9 : Printf (" Ninty");
break;
```

```
}
if (r[2] != 1)
{
    switch (r[1])
    {
        case 1: Printf ("One");
        break;
        Case 2: Printf ("Two");
        break;
        Case 3: Printf ("Three");
        break;
        Case 4: Printf ("Four");
        break;
        Case 5: Printf ("Five");
        break;
        Case 6: Printf ("Six");
        break;
        Case 7: Printf ("Seven");
        break;
        Case 8: Printf ("Eight");
        break;
        Case 9: Printf ("Nine");
    } break;
}
Printf ("Rupees only");
}
```

Lab 4: 1D Array manipulation:

a) Write a C program for the following Operations on a N array.

1) Insert at a Position:-

```
#include <stdio.h>
Void main()
{
int i, a[10], n, ele, pos;
Printf("enter the size of array\n");
scanf("%d", &n);
Printf("enter the elements in array\n");
for(i=0; i<n; i++)
scanf("%d", &a[i]);
Printf("enter the element you coant to insert\n");
Scanf("%d", &ele);
Printf("enter the position\n");
Scanf("%d", &pos);
for(i=n-1; i>=pos; i--)
a[i+1]= a[i];
a[pos] = ele;
n++;
for(i=0; i<n; i++)
Printf("%d\n", a[i]);
}
```

ii) Delete an element:-

```
#include <stdio.h>
Void main()
{
  int i, a[10], n, pos;
  printf("enter the size of array \n");
  scanf("%d", &n);
  printf("enter the elements in array \n");
  for(i=0; i<n; i++)
  scanf("%d", &a[i]);
  printf("enter the position of element you want to delete \n");
  scanf("%d", &pos);
  for(i=pos; i<n-1; i++)
    a[i] = a[i+1];
    n--;
    printf("after deleting \n");
    for(i=0; i<n; i++)
    printf("%d", a[i]);
}
```

Output 1:

```
enter the size of array
5
enter the elements in array
55
45
30
80
90
enter the position of element you want to delete
3
after deleting
55
45
30
90
```

Output 2:

```
enter the size of array
4
enter the elements in array
35
60
80
75
enter the position of element you want to delete
0
after deleting
60
80
75
```

(iii) Find the sum of all elements :-

```
#include <stdio.h>
void main()
{
    int i, a[10], sum=0, n;
    printf("enter size of array \n"); scanf("%d", &n);
    printf("enter elements of array \n");
    for(i=0; i<n; i++)
    {
    scanf("%d", &a[i]);
    sum = sum + a[i];
    }
    printf("sum of array elements is %d\n", sum);
}
```

---

output 1:

```
enter size of array
5
enter elements of array
15
20
10
8
5
sum of array elements is 58
```

---

output 2:

```
enter size of array
4
enter elements of array
-60
40
30
20
Sum of array elements is 30
```

(iv) Count of positive, negative, odd and even numbers

```
#include <stdio.h>
Void main()
{
  int i, n, K, a[10], c=0, g=0, S1=0, S2=0;
  Printf("enter size of array \n");
  scanf("%od", &n);
  printf("enter array elements \n");
  for(i=0; i<n; i++)
    scanf("%od", &a[i]);
  Printf("enter 1 to get count of positive, negative numbers,
          2 to get odd and even count \n");
  scanf("%od", &K);
  Switch(K)
  {
    case 1: for(i=0; i<n; i++)
            {
              if(a[i]>0)
                c++;
              else
                g++;
            }
            Printf("no. of positive numbers = %od \n", c);
            printf("no. of negative numbers = %od \n", g);
    break;
```

```
case 2: for (i=0; i<n; i++)
        {
          if (a[i] % 2 == 0)
          s1= s1+ a[i];
          else
          s2 = s2+ a[i];
        }
        Printf ("No. of even numbers is %d \n", s1);
        printf ("No. of odd numbers is %od \n", s2);
      break;
      default : printf ("enter 1 or 2 \n");
      break;
    }
}
```

Output 1:

enter size of array

4

enter array elements

4

-5

4 4

3 0

enter 1 to get count of positive, negative numbers, 2 to get odd and even count

1

no. of positive numbers = 3

no of negative numbers = 1

Output 2:

enter size of array

4

enter array elements

4

5

8

3

enter 1 to get count of positive, negative numbers, 2 to get odd and even count

2

No. of even numbers is 2

No of odd numbers is 2

(v) Search for an element:-

```
#include <stdio.h>
Void main()
{
  int a[10], n, ele, low, high, mid, pos, i, flag;
  printf("enter the size of array\n");
  scanf("%d", &n);
  printf("enter elements in ascending order\n");
  for(i=0; i<n; i++)
  scanf("%d", &a[i]);
  printf("enter the element to be searched\n");
  scanf("%d", &ele);
  low=0, high=n-1, flag=0;
  while(low<=high)
  {
    mid=(low+high)/2;
    if(ele==a[mid])
    {
      flag=1;
      pos=mid+1;
      break;
    }
    else if(ele>a[mid])
    low=mid+1;
    else
    high=mid-1;
  }
```

```
if (flag==1)
printf("The element is found at %d position \n", pos);
else
printf("The element is not found \n");
}
```

Output 1:

```
enter the size of array
5
enter elements in ascending order
5
15
20
25
80
enter the element to be searched
25
The element is found at 4 position
```

Output 2:

```
enter the size of array
4
enter elements in ascending order
6
16
24
80
enter the element to be searched
16
The element is found at 2 position
```

vi) Interchange two position values:

```
#include <stdio.h>
void main()
{
int n, a[10], pos1, pos2, t;
printf("enter the size of array \n");
scanf("%d", &n);
printf("enter elements of array \n");
for(i=0; i<n; i++)
scanf("%d", &a[i]);
printf("enter two index positions to interchange their
        values \n");
scanf("%d %d", &pos1, &pos2);
t = a[pos1]; a[pos1] = a[pos2]; a[pos2] = t;
printf("after interchanging \n");
for(i=0; i<n; i++)
printf("%d", &a[i]);
```

Output1:

```
enter the size of array
5
enter elements of array
4
5
5 5
6
8
enter two index positions to interchange their values
1
3
after interchanging
4
6
5 5
5
3
```

Output 2:

```
enter the size of array
6
enter elements of array
4
5
80
19
15
20
enter two index positions to interchange their values
2
5
after interchanging
4
5
20
19
15
80
```

ab5:- Problems on 2D arrays and Strings

Write a C program to

Read a Matrix, and to find the transpose of the matrix

Addition of two matrices

Multiplication of two matrices

Read a Matrix and to find transpose of the matrix:-

```
#include <stdio.h>
void main()
{
int a[10][10], b[10][10], i, j, m, n;
printf("Enter the order of matrix \n");
scanf("%d%d", &m, &n);
Printf("Enter matrix elements\n");
for(i=0; i<m; i++)
for(j=0; j<n; j++)
scanf("%d", &a[i][j]);
for(i=0; i<n; i++)
for(j=0; j<m; j++)
b[i][j]=a[j][i];
Printf("\n The transpose of matrix A is \n");
```

```
for(i=0; i<n; i++)
{
  for(j=0; j<m; j++)
    Printf("%d\t", b[i][j]);
  }
  Printf("\n");
}
```

Output:-

```
Enter the order of matrix
3
2
Enter matrix elements
5
6
7
2
3
4
```

$$\begin{bmatrix} 5 & 6 \\ 7 & 2 \\ 3 & 4 \end{bmatrix}$$

The transpose of matrix A is

```
5   7   3
6   2   4
```

(ii) Addition of two Matrices :-

```
#include <stdio.h>
void main()
{
  int a[10][10], b[10][10], c[10][10], i, j;
  int m, n, P, q;
  printf("Enter the order of first matrix \n");
  scanf("%d%d", &m, &n);
  printf("Enter first matrix elements \n");
```

```
for(i=0; i<m; i++)
for(j=0; j<n; j++)
scanf("%d", &a[i][j]);
printf("Enter the order of second matrix\n");
scanf("%d%d", &p, &q);
printf("Enter second matrix elements\n");
for(i=0; i<p; i++)
for(j=0; j<q; j++)
scanf("%d", &b[i][j]);
if(m==p&&n==q)
{
  for(i=0; i<m; i++)
   for(j=0; j<n; j++)
    c[i][j]= a[i][j]+ b[i][j];
  printf("\nThe resultant matrix after addition\n");
  for(i=0; i<m; i++)
  {
   for(j=0; j<n; j++)
    printf("%d\t", c[i][j]);
  }
  printf("\n");
}
else
printf("Matrix addition is not possible. Order
        should be same for both matrices\n");
}
```

Output:-

```
Enter the order of first matrix
2
2
Enter first materix elements
4
5
1
2
Enter the order of second matrix
2
2
Enter second matrix elements
2
2
3
4

The resultant matrix after addition
6   7
4   6
```

Program:-

iii) Multiplication of two matrices with prompting of rules

```
#include <stdio.h>
void main()
{
    int a[10][10], b[10][10], c[10][10], i, j, k, m, n, p;
    printf("enter the first matrix order\n");
    scanf("%d%d", &m, &n);
```

```
printf("enter the first matrix elements\n");
for(i=0; i<m; i++)
for(j=0; j<n; j++)
scanf("%d", &a[i][j]);
printf("enter second matrix order\n");
scanf("%d %d", &p, &q);
printf("enter second matrix elements\n");
for(i=0; i<p; i++)
for(j=0; j<q; j++)
scanf("%d", &b[i][j]);
if(n==p)
{
/* Resultant matrix will be of order (no. of rows in
1st matrix x no. of columns in second matrix)*/
for(i=0; i<m; i++)
for(j=0; j<q; j++)
{
c[i][j]=0;
for(k=0; k<n; k++)
/* k<n or k<p as n=p */
c[i][j]= c[i][j]+ (a[i][k] * b[k][j]);
}
Printf("\n Resultant matrix after matrix
multiplication\n");
for(i=0; i<m; i++)
for(j=0; j<q; j++)
{
Printf("%d\t", c[i][j]);
```

```
Printf("\n");
}
else
printf("\nFor matrix multiplication, The column order
    of first matrix should be equal to row order
    of second matrix\n");
}
```

Output:

```
enter the first matrix order
2
2
enter the first matrix elements
2
3
4
6
enter the second matrix order
2
2
enter the second matrix elements
3
2
1
1
```

$$\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

Resultant matrix after matrix multiplication

```
 9   7
18  14
```

⑥ Write a C program that uses functions to perform the following operations

i) To insert a substring into a given mainstring from a given position

```
#include <stdio.h>
void insert (char [ ]);
void main( )
{
  char S1[100];
  printf("Enter the main string \n");
  scanf("%[^\n]", S1);
  insert(S1);
}
void insert (char ch[100])
{
  char S2[20];
  int i, j, p, l1, l2;
  printf("Enter the substring \n");
  scanf("%s", S2);
  printf("Enter position in main staing you want to
        insert \n");
  scanf("%d", &p);
  for(l1=0; ch[l1]!='\0'; l1++);
  for(l2=0; S2[l2]!='\0'; l2++);
  ch[l1+l2]='\0';
  for(i=l1+l2-1, j=l1-1; j>=p-1; i--, j--)
  ch[i]=ch[j];
  for(i=p-1, j=0; j<l2; i++, j++)
```

```
    ch[i] = s2[j];
    Printf("\n After inserting, the main string is \n");
    Printf("%s\n", ch);
}
```

Output 1:-

Enter the main string
Kohli is cricketer
Enter the substring
great
Enter position in main string you want to insert
10

After inserting, the main string is

kohli is great cricketer.

Output 2:-

Enter the main string
Wonderful
Enter the substring
o
Enter position in main string you want to insert
2
After inserting, the main string is

wonderful

(ii) To delete n characters from a given position in a given string

```
#include <stdio.h>
void insert(char [ ]);
void main()
{
    char s1[100];
    printf("Enter the string\n");
    scanf("%[^\n]", s1);
    insert(s1);
}
void insert(char ch[100])
{
    int i, j, p, n;
    printf("Enter position and no. of characters you
            want to delete\n");
    scanf("%d%d", &p, &n);
    for(i=p-1, j=p+n-1; i<p+n; i++, j++)
    ch[i]=ch[j];
    ch[j]='\0';
    printf("\n After deleting, the string is\n\n");
    printf("%s\n", ch);
}
```

Output 1:-

Enter the string
hello hai
Enter position and no. of characters you want to delete
2
4
After deleting, the string is

h hai

Output 2:-

Enter the string
rajeshKumar
Enter position and no. of characters you want to delete
7
3

After deleting, the string is

rajeshars

C. Write a C program to find string within a sentence and replace it with another string.

```
#include <stdio.h>
# include <string.h>
void rs (char []);
void main()
{
    char z[100];
    printf ("Enter a sentence \n");
    scanf ("%[^\n]", z);
    rs (z);
}
Void rs (char ch[100])
{
    char s[20], r[20];
    int l1, l2, l3, i, j, k, p, q, t, flag=0;
    Printf ("Enter the substring you want to remove
```

```
scanf("%s", s);
l1 = strlen(ch), l2 = strlen(s);
for(i=0; ch[i] != '\0'; i++)
{
  k=0;
  for(j=i; j<i+l2; j++)
  {
    if (ch[j] == s[k])
    k++;
  }
  if (k == l2)
  {
    p=t=i, q=j;
    flag=1;
    break;
  }
}
if (flag==0)
printf("Entered substring not found \n");
else
{
  while (q<=l1)
  {
    ch[p] = ch[q];
    p++;
    q++;
  }
  printf("Enter substring you want to replace in \n");
  scanf("%s", r);
```

```
    l3 = strlen(r);
    while(p>=t)
    {
      ch[p+l3] = ch[p];
      p--;
    }
    for(v=0; r[v]!='\0'; v++, t++)
    ch[t] = r[v];
    printf("\n The resultant string is \n\n");
    puts(ch);
    }
}
```

Output 1 :-

Enter a sentence
Rahul is a good boy
Enter the substring you want to remove
good
Enter string you want to replace in
bad
The resultant string is
Rahul is a bad boy

Output 2 :-

Enter a sentence
Rohit is excellent
Enter the substring you want to remove
Rohit
Enter string you want to replace in
Rohini
The resultant string is
Rohini is excellent

5.d: Write a C program that reads a line of text and counts all occurance of a particular coord

```
#include <stdio.h>
#include <string.h>
void cs(char[]);
void main()
{
    char z[100];
    printf("Enter a sentence\n");
    scanf("%[^\n]", z);
    cs(z);
}
void cs(char ch[100])
{
    char s[20];
    int l1, l2, i, j, k, c=0;
    printf("Enter the coord\n");
    scanf("%s", s);
    l1=strlen(ch), l2=strlen(s);
    for(i=0; ch[i]!='\0'; i++)
    {
        k=0;
        for(j=i; j<i+l2; j++)
        {
            if(ch[j]==s[k])
            k++;
        }
        if(k==l2)
        c++;
    }
```

```
if (c==0)
printf ("\n Entered word not found\n");
else
{
  printf ("\n The given word occured %d times\n", c);
}
}
```

Output 1:-

Enter a sentence
johny johny yes papa
Enter the word
johny

The given word occured 2 times.

Output 2:-

Enter a sentence
He studies well. He is topper as well.
Enter the word
well

The given word occured 2 times.

5.e:- Ram wanted to increase his typing speed, to participate in programming contests.
A sentence is known as pangram if it contains every letter of the alphabet.
Read a sentence from user, store it in a character array 'S'. Hint: Allocate memory for the string using dynamic memory allocation and determine whether the given string is a pangram or not. Ignore upper or lower cases.

```
#include <stdio.h>
#include <ctype.h>
void main()
{
  char s[100], j;
  int x, y, k;
  printf("Enter sentence\n");
  scanf("%[^\n]", s);
  x=0;
  for(j=65; j<=90; j++)
  {
    y=0;
    for(k=0; s[k]!='\0'; k++)
    {
      if(j==toupper(s[k]))
      y++;
    }
    if(y>0)
    x++;
  }
  if(x==26)
  printf("\n The given sentence is pangram\n");
else
  printf("\n The given sentence is not pangram\n");
```

Allocating Dynamically

```
char *s;
s=(char*)malloc
  (10*sizeof(char));
```

Remaining program same

Output 1:

```
Enter sentence
qwerty

The given sentence is not pangram
```

Output 2:

```
Enter sentence
The quick brown fox jumps over
the lazy dog

The given sentence is Pangram
```

Lab6: Function calling mechanisms: (call by Value)

a) Write a C program to find prime Fibonacci numbers using functions

```
# include <stdio.h>
void pf(int);
void main()
{
    int n;
    printf("Enter no. of prime number elements for in
            fibonacci series\n");
    scanf("%d", &n);
    pf(n);
}
void pf(int b)
{
    int a[50], i, c, j;
    a[0]=0, a[1]=1;
    for(i=2; i<2*(b+1); i++) //out of 2(b+1)
numbers, b prime numbers will be printed//
    a[i]= a[i-1] + a[i-2];
    Printf("\n The prime fibonacci series is \n");
    for(i=0; i<2*(b+1); i++)
    {
        c=0;
        for(j=1; j<=a[i]; j++)
```

```
{  if (a[i]%j==0)
     c++;
  {
  if (c==2&&d<=b)
   {  printf("%d,", a[i]);
      d++;
   }
  }
}
```

Output 1 :-

Enter no. of prime number elements for in fibonacci series

The prime fibonacci series is

2, 3, 5,

Output 2 :

Enter no. of prime number elements for in fibonacci series

The prime fibonacci series is

2, 3, 5, 13, 89, 233,

b) Write a C program to find npr, ncr values by writing factorial function

```
#include <stdio.h>
void npcr(int, int);
void main()
{
    int n, r;
    printf("Enter n and r values\n");
    scanf("%d%d", &n, &r);
    npcr(n, r);
}
void npcr(int m, int p)
{
    int i;
    float f1, f2, f3;
    f1 = f2 = f3 = 1;
    for (i=1; i<=m; i++)
    f1 = f1 * i;
    for (i=1; i<=p; i++)
    f3 = f3 * i;
    for (i=1; i<=m-p; i++)
    f2 = f2 * i;
    printf("The values of npr and ncr are
            %f and %f\n", f1/f2, f1/(f2*f3));
}
```

Output:

```
Enter n and r values
5
2
The values of npr and ncr are 20.000000 and
10.000000
```

output 2:

Enter n and r values

6

3

The values of npr and ncr are 120.000000

and 20.000000

Lab 7:- Function calling mechanisms (call by reference)

a) Write a C program to perform all operations on strings using functions

i) Find the length of a string

```
#include <stdio.h>
void str(char *);
void main()
{
    char s[20];
    printf("Enter a word\n");
    scanf("%[^\n]", s);
    str(s);
}
void str(char *c)
{
    int l=0;
    while(*c!='\0')
    {
        l++;
        c++;
    }
    printf("The length of string is %d\n", l);
}
```

Output 1
Enter a word
success
The length of string is 7

Output 2
Enter a coord
optimistic
The length of string is 10

ii) Find the substring when position and length is given.

```
#include <stdio.h>
void str(char *);
void main()
{
  char s[100];
  printf("Enter a string \n");
  scanf("%0 [^\n]", s);
  str(s);
}
void str(char *c)
{
  int p, l, i;
  printf("Enter position and length of substring \n");
  scanf("%od %od", &p, &l);
  printf("\n Required substring is ");
  for(i=p-1; i<p-1+l; i++)
  printf("%c", c[i]);
}
```

Output 1
Enter a string
My name is sunil
Enter position and length of substring
4
4
Required substring is name

Output 2
Enter a string
Beautiful
Enter position and length of substring
7
3
Required substring is ful

Lab 8 Recursive functions

a) Write a C programs that use recursive function

i) To find the factorial of a given integer:

```
#include <stdio.h>
int fact(int);
int main()
{
    int n, f;
    printf("Enter the number \n");
    scanf("%d", &n);
    f = fact(n);
    printf("\n The factorial of given number is %d \n", f);
}
int fact(int n)
{
    int f;
    if (n==0 || n==1)
    f=1;
    else
    f = n * fact(n-1);
    return f;
}
```

Output1

Enter the number
5
The factorial of given number is 120.

Output2

Enter the number
8
The factorial of given number is 40320

iii) To find the greatest common divisor of two given integers

```
#include <stdio.h>
int gcd (int, int);
int main()
{
  int a, b, c;
  printf("Enter two numbers\n");
  scanf("%d %d", &a, &b);
  c = gcd(a, b);
  printf("\n The greatest common divisor of two
          numbers is %d\n", c);
}
int gcd (int a, int b)
{
  int c;
  if(b==0)
  c=a;
  else
  c=gcd(b, a%b);
  return c;
}
```

Output1

Enter two numbers
7
2
The greatest common divisor of two numbers is 1

Output2

Enter two numbers
8
4
The greatest common divisor of two numbers is 4.

iii) To print Fibonacci series :-

```
#include <stdio.h>
int fib(int);
int main()
{
    int n, i;
    Printf("Enter how many numbers\n");
    scanf("%d", &n);
    printf("\nThe fibonacci numbers are\n");
    Printf("%d\t%d", 0, 1);
    for(i=1; i<n-1; i++)
    Printf("\t%d", fib(i));
}
int fib(int n)
{
    int c;
    if (n==0 || n==1)
    c=1;
    else
    c = fib(n-1) + fib(n-2);
    return c;
}
```

Output:-

Enter how many numbers

8

The fiboracci numbers are

0 1 1 2 3 5 8 13

Lab 9 Dynamic memory allocation

a) Write a C program to concatenate two strings using pointers

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
char *c, *s;
int i, j;
c=(char*)malloc(10*sizeof(char));
s=(char*)calloc(1, sizeof(char));
printf("Enter 1st string \n");
scanf("%[^\n]", c);
printf("Enter 2nd string \n");
scanf("%s", s);
for(j=0; c[i]!='\0'; i++);
for(j=0; c[j]!='\0'; j++, i++)
c[i]=s[j];
Printf("\n After concatenating two strings,\n\n");
Printf("%s", c);
}
```

Output:

```
Enter 1st string
RVRJCCE College
Enter 2nd string
Guntur
After concatenating two strings
RVRJCCE CollegeGuntur
```

b) Write a C program to find the length of string using Pointers

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
  char *s;
  s=(char *)malloc(sizeof(char));
  printf("Enter string\n");
  scanf("%0[^\n]", s);
  int l=0;
  while(*s!='\0')
  {
    l++;
    s++;
  }
  printf("The length of string is %d\n", l);
}
```

output1:

```
Enter string
RVRJCCE College Bunter
The length of storing is 22.
```

output2:

```
Enter string
Computer Science
The length of string is 16
```

5) Write a C program to compare two strings using pointers

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  char *c, *s;
  int k=0;
  c=(char*) malloc (10*sizeof(char));
  s=(char*) calloc (1, sizeof(char));
  printf("Enter 1st string \n");
  scanf("%s", c);
  printf("Enter 2nd string \n");
  scanf("%s", s);
  while(*c!='\0' || *s!='\0')
  {
    if(*c!=*s)
    {
      k=1;
      break;
    }
    else
    {
      c++;
      s++;
    }
  }
  if(k==0)
  printf("\n Strings are same \n");
  else
  printf("\n Strings are not same \n");
}
```

Output 1:

Enter 1st string
college
Enter 2nd string
college

Strings are same.

Output 2:

Enter 1st string
Mangoes
Enter 2nd string
Mahi

Strings are not same.

---

d) Write a C program to copy a string from source to destination using pointers

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char *s, *d;
    int i, j;
    s = (char *) malloc (10 * sizeof (char));
    d = (char *) calloc (1, sizeof (char));
    printf ("Enter string in source\n");
    scanf ("%[^\n]", s);
    for (i=0, j=0; s[i]!='\0'; i++, j++)
    d[j] = s[i];
    Printf ("source text is %s\n", s);
    Printf ("destination text is %s\n", d);
}
```

Output:-

Enter string in source
Programming for problem solving
source text is programming for problem solving
destination text is Programming for problem solving.

Lab 10: Structures and Unions:

a) Create a Book structure containing book-id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.

```
#include <stdio.h>
struct book
{
    int id, price;
    char title[20], author[20];
};
void printb(struct book a);
void main()
{
    struct book a;
    printf("Enter the book id \n");
    scanf("%d", &a.id);
    printf("Enter the price of book\n");
    scanf("%d", &a.price);
    printf("Enter title of book\n");
    scanf("%s", a.title);
    printf("Enter author name\n");
    scanf("%s", a.author);
    printb(a);
}
void printb(struct book b)
{
    printf("\n The book id number is %d \n", b.id);
```

```
printf("The price of the book is %d\n", b.price);
printf("The title of the book is %s\n", b.title);
printf("The author name is %s\n", b.author);
}
```

Output:-

```
Enter the book id
12345678
Enter the price of book
1200
Enter title of book
PROGRAMMING
Enter author name
S.KUMAR

The book id number is 12345678
The price of the book is 1200
The title of the book is PROGRAMMING
The author name is S.KUMAR.
```

(b). Create a Union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address

```
#include <stdio.h>
#include <string.h>
union adr
{
    char name[30], hoa[20], hsa[20], city[15],
    state[15];
    int zip;
};
```

```
void main()
{ union adr a;
  printf("My address is \n\n");
  strcpy(a.name, "D.K. Sunil Kumar Reddy");
  printf("Name : %s \n", a.name);
  strcpy (a.hoa, "Giddalur");
  printf("Home address : %s \n", a.hoa);
  strcpy (a.hsa, "Guntur");
  printf ("Hostel address : %s \n", a.hsa);
  strcpy(a.city, "Ongole");
  printf("City : %s\n", a.city);
  strcpy (a.state, "Andhra Pradesh");
  printf("State : %s \n", a.state);
  a.zip = 123456;
  printf("Zipcode : %d \n", a.zip);
}
```

Output:-

My address is

Name : D.K. Sunil Kumar Reddy
Home address : Giddalur
Hostel address : Guntur
City : Ongole
State : Andhra Pradesh
Zipcode : 123456

2. Write C program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures, display the name and date of birth

```
#include <stdio.h>
struct name
{
  char name[30];
};
struct DOB
{
   struct name c;
   int date,month, year;
}a;
int main()
{
 printf("Enter your name\n");
 scanf("%[^\n]", a.c.name);
 printf("Enter date, month no., year of your birthday\n");
 scanf("%d%d%d", &a.date, &a.month, &a.year);
 printf("\nName : %s\n Date of birthday : %d/%d/%
        \n", a.c.name, a.date, a.month, a.year);
}
```

Output:

```
Enter your name
Santosh
Enter date, month no., year of your birthday
25
12
2008
```

Name: Santosh
Date of birthday: 25/12/2008

Lab11: File Operations

a) Write a C program to display the contents of a file

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
  FILE *fp;
  char ch, filename[15];
  fp=fopen("abcd.txt","a");
  if(fp==NULL)
  {
  printf("Enter the filename to be opened\n");
  scanf("%s", filename);
  fp=fopen(filename, "a");
  if(fp==NULL)
  {
    printf("Unable to open file\n");
    exit(0);
  }
  printf("Enter some text to add\n");
  ch=getchar();
  while(ch!=EOF)
  {
    fputc(ch, fp);
    ch=getchar();
  }
  fclose(fp);
```

```
fp = fopen (filename, "r");
printf ("\n The contents of file are \n");
  ch = fgetc(fp);
  while (ch != EOF)
  {
    printf ("%c", ch);
    ch = fgetc (fp);
  }
  fclose (fp);
}
```

Output:

```
Enter the filename to be opened
abcd.txt
Enter some text to add
My name is sunil

The contents of file are
Hello hi to everyone
My name is sunil
```

file in append mode

file in read mode

---

b) Write a C program to copy the contents of one file to another

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char S[15], d[15], ch;
    FILE *fp1, *fp2;
```

```
printf("Enter the source filename \n");
scanf("%s", s);
printf("Enter the filename where to be copied \n");
scanf("%s", d);
fp1 = fopen(s, "r");
if (fp1 == NULL)
{ printf("Unable to open file \n");
  exit(0);
}
fp2 = fopen(d, "a");
if (fp2 == NULL)
{ printf("Unable to open file \n");
  exit(0);
}
ch = fgetc(fp1);
while (ch != EOF)
{ fputc(ch, fp2);
  ch = fgetc(fp1);
}
fclose(fp1);
fclose(fp2);
}
```

Output :-

Enter the source filename
a.txt
Enter the filename where to be copied
b.txt
/* open respective files to check content is copied or not */

6) Write a C program to reverse the first n characters in a file, where n is given by the user.

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    FILE *fp1, *fp2;
    char ch;
    int n;
    fp1 = fopen("a.txt", "w+");
    if (fp1 == NULL)
    {
        printf("Unable to open file\n");
        exit(0);
    }
    printf("Enter text into file\n");
    ch = getchar();
    while (ch != EOF)
    {
        fputc(ch, fp1);
        ch = getchar();
    }
    printf("Enter no. of characters to be replaced\n");
    scanf("%d", &n);
    fp2 = fopen("r.txt", "w");
    fseek(fp1, n+1, 0);
    while (fseek(fp1, -2, 1) == 0)
    {
        ch = fgetc(fp1);
        fputc(ch, fp2);
    }
```

```
fseek (fp, n, 0);
ch = fgetc (fp1);
while (ch != EOF)
{ fputc (ch, fp2);
  ch = fgetc (fp1);
}
fclose (fp1);
fclose (fp2);
}
```

output:-

Enter text into file
good morning
Enter no. of characters to be replaced
6

Contents in r.txt when we opened file,
m doogoorning

---

d) Two files DATA1 and DATA2 contain sorted list of integers. Write a C program to merge the contents of two file into a third file DATA

```
#include <stdio.h>
int main(int argc, char *argv[])
{
  FILE *fp, *fp2;
  char ch;
  int i;
  fp2 = fopen("DATA.txt", "a");
```

```
for (i=1; i<argc; i++)
{
    fp=fopen (argv[i], "r");
    ch = fgetc (fp);
    while (ch != EOF)
    {
        fputc (ch, fp2);
        ch = fgetc (fp);
    }
}
fclose (fp);
fclose (fp2);
}
```

```
$. /a.out  DATA1.txt  DATA2.txt
```

// open DATA.txt to check whether merged or not. Before writing program, create files DATA1 and DATA2 which contains sorted lists of integers //

---

e. Write a C program to count no. of characters present in the file

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char filename[15], ch;
    FILE *fp;
    int c, s;
```

```
printf("Enter an [redacted] filename\n");
scanf("%s", filename);
fp = fopen(filename, "w+");
printf("Enter some text to file\n");
ch = getchar();
while(ch != EOF)
{ fputc(ch, fp);
  ch = getchar();
}
fseek(fp, 2, 0);
c = 0; s = 0;
ch = fgetc(fp);
while(ch != EOF)
{ c++;
  if(ch == ' ')
  s++;
  ch = fgetc(fp);
}
fclose(fp);
Printf("\n No. of characters in given file = %d\n", c);
 Printf("No. of characters without including
        spaces = %d\n", c-s);
}
```

Output:

Enter an filename
c:txt
Enter some text to file
C Programming for Problem solving

No. of characters in given file = 33
No. of characters without including spaces = 29.

---

8. Write a C program to find whether the given word is present in the given file or not.

```
#include <stdio.h>
#include <string.h>
int main()
{
    FILE *fp;
    char t[15], w[15], file[15];
    int ch, c=0;
    printf("Enter file name \n");
    scanf("%s", file);
    fp=fopen(file, "w");
    printf("\n Enter the text \n");
    ch=getchar();
    while(ch, fp);
    {
    ~~ch=getchar();~~
     fputc(ch, fp);
     ch=getchar();
    }
    printf("\n Enter the coord to be searched \n");
    scanf("%s", w);
    fclose(fp);
    fp=fopen(file, "r");
```

```
while ( !feof (fp))
{ fscanf(fp, "%s", t);
  if (strcmp(t, w) == 0)
  c++;
}
fclose(fp);
if (c == 0)
printf("Given coord not found \n");
else
printf("\n The coord %s is occured %d times",
        w, c);
}
```

Output1:-

Enter file name
e.txt

Enter the text
He studies well. He is topper as well

Enter the coord to be searched
well

The coord well is occured 2 times.

Output2:- Enter file name
fo.txt
Enter the text
Old is gold.
Enter the coord to be searched
new
Given coord not found.

Lab 12:- Command line arguments

g) Write a C program to print all arguments given through command line

```
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[])
{
  Printf ("\n Number of arguments supplied = %d \n", argc);
  if (argc < 1)
  {
    printf ("\n Insufficient number of arguments");
    exit(0);
  }
  Printf("\n The arguments supplied are \n");
  for (i=0; i< argc; i++)
  Printf("%s\t", argv[i]);
}
```

Output

```
$ cc com.c
$ /a.out apple red healthy good
Number of arguments supplied = 5
The arguments supplied are: a.out apple red healthy
  good
```

b) Write a program to add numbers using command line arguments

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
  int sum, i;
  for(i=1, sum=0; i<argc; i++)
    sum = sum + atoi(argv[i]);
  printf("\n sum=%d", sum);
}
```

Output:-

```
$cc. sume.c
$.a.out 12 43 65 34
sum=154
```