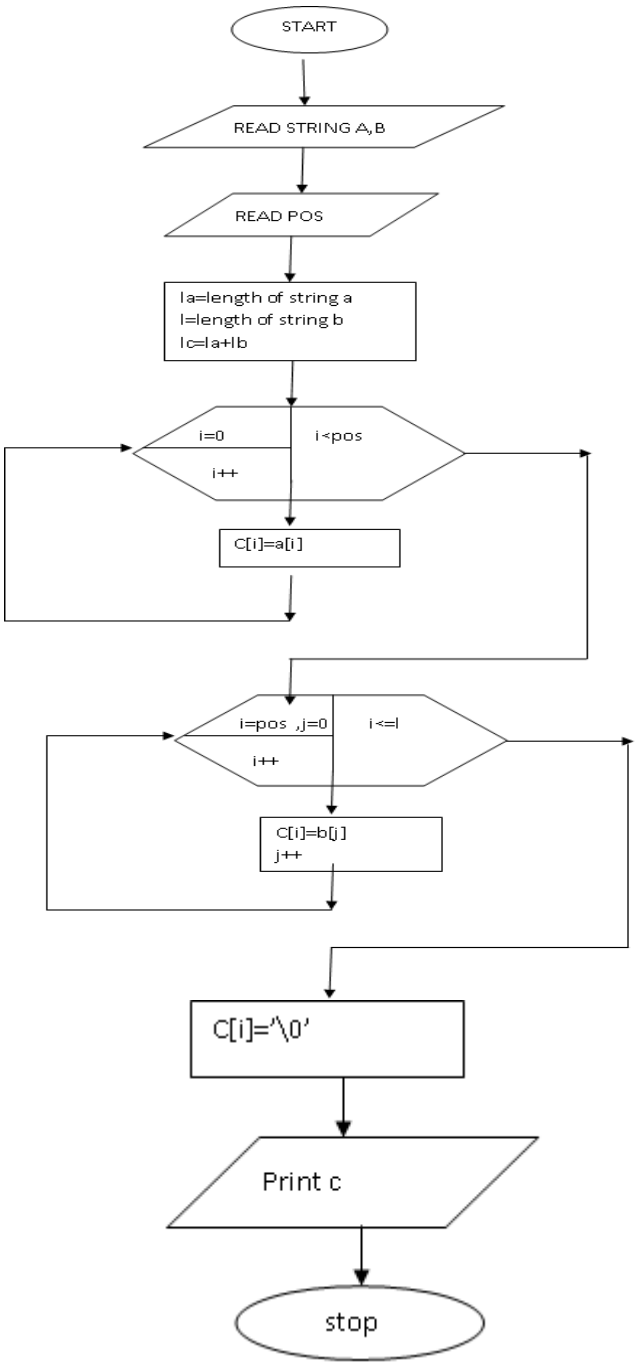
Week: 10

10) a) Write a C program to use function to insert a sub-string in to given main string from a given position.

Aim:

To insert a string into another string from a specified position.

Flow Chart :



Algorithm:

Step 1: start
Step 2: read main string and sub string
Step 3: find the length of main string(r)
Step 4: find length of sub string(n)
Step 5: copy main string into sub string
Step 6: read the position to insert the sub string(p)
Step 7: copy sub string into main string from position p-1
Step 8: copy temporary string into main string from position p+n-1
Step 9: print the strings
Step 10: stop

Program:

```
#include<stdio.h>
#include<string.h>
main()
{
char a[3qq0],b[30],c[30];
int pos=0,i=0,l,la,lb,lc,j;
        puts("Enter a string");
        gets(a);
        puts("Enter sub string");
        gets(b);
        puts("enter position for insertion");
        scanf("%d",&pos);
        la=strlen(a);
        lb=strlen(b);
        l=pos+lb;
        lc=la+lb;
        for(i=0;i<pos;i++)
        {
                c[i]=a[i];
        }
        j=0;
        for(i=pos;i<=l;i++)
        {
                c[i]=b[j];
                j++;
        }
        j=pos;
        for(i=l;i<lc;i++)
        {
                c[i]=a[j];
                j++;
        }
        c[i]='\0';
        puts("String after Insertion is:");
        printf("%s",c);
```

```
}
```

Input:

```
Enter First String:
  Comer
Enter Second String:
   put
```

Output:

```
Enter the position where the item has to be inserted:3

  Computer
```

Record at least 3 results

Signature of faculty with date

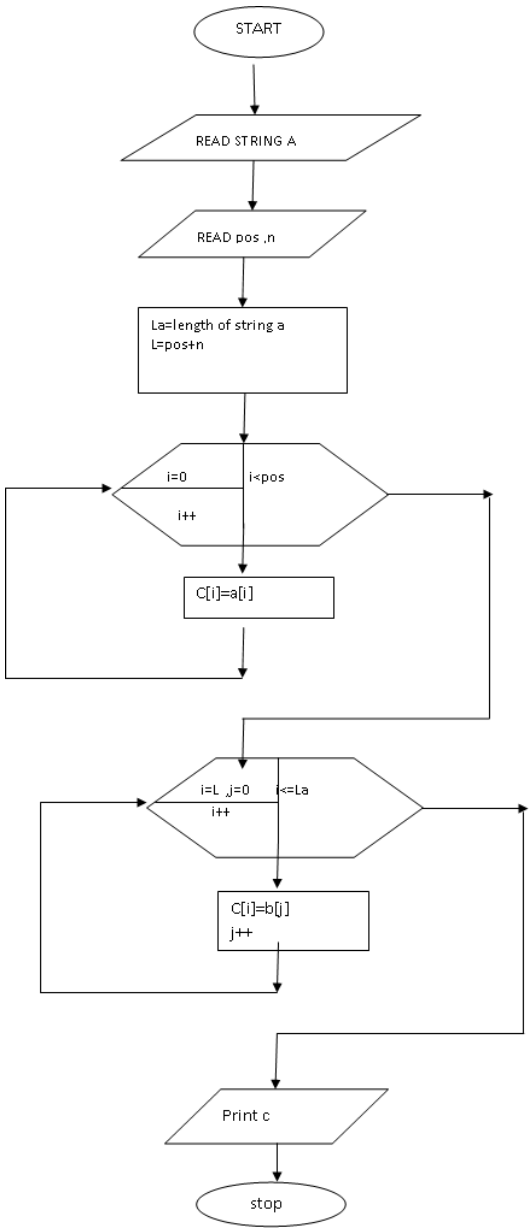
10) b) To delete n Characters from a given position in a given string.

Aim: To delete n Characters from a given position in a given string.

Algorithm:

Step 1: start
Step 2: read string
Step 3: find the length of the string
Step 4: read the value of number of characters to be deleted and positioned
Step 5: string copy part of string from position to end, and
(position + number of characters to end)
Step 6: stop

Flow Chart:



Program:

```
#include<stdio.h>
        #include<string.h>
        main()
        {
        char a[30],c[30];
        int pos=0,i=0,L,La,j,n;
                puts("Enter a string");
                gets(a);
                puts("enter position  for deletion");
                scanf("%d",&pos);
                puts("Enter number of characters to be deleted");
                scanf("%d",&n);
                La=strlen(a);
                L=pos+n;
                for(i=0;i<pos;i++)
                {
                        c[i]=a[i];
                }
                j=pos;
                for(i=L;i<=La;i++)
                {
                        c[j]=a[i];
                        j++;
                }
                puts("String after Deletion is:");
                printf("%s",c);
 }
```

Input:

Enter the string
jayapal

Enter the position from where to delete:4
Enter the number of characters to be deleted 2

Output:

jayal

Record at least 3 results

Signature of faculty with date

Week: 11

11) a) Write a C program using user defined functions to determine whether the given string is palindrome or not.

Aim: To determine if the given string is palindrome or not.

Description :

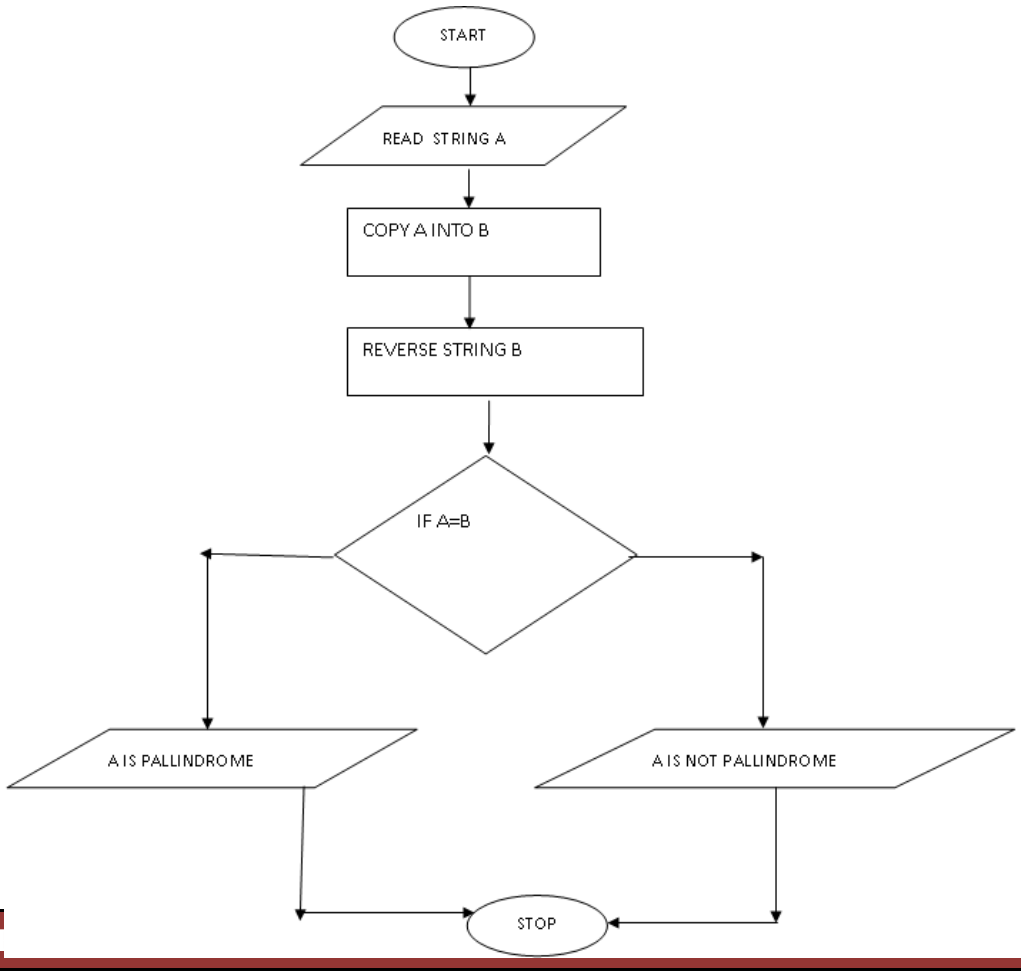
Palindrome means string on reversal should be same as original

Ex: madam on reversal is also madam

Algorithm:

Step 1: start
Step 2: read string A
Step 3: copy string A into B
Step 4: reverse string B
Step 5: compare A &B
If A equals B to got step 6
Else goto step 7
Step 6:print given string A is pallindrom
Step 7:print given string is not pallindroma
Step 8: stop

Flow Chart:



Program:

```
#include <stdio.h>
#include <string.h>
 void main()
{
  char string[25], reverse_string[25] = {'\0'};
   int i, length = 0, flag = 0;

   printf("Enter a string \n");
   gets(string);
   for (i = 0; string[i] != '\0'; i++)
   {
      length++;
   }
  printf("The length of the string '%s' = %d\n", string, length);
   for (i = length - 1; i >= 0 ; i--)
  {
      reverse_string[length - i - 1] = string[i];
  }

  for (flag = 1, i = 0; i < length ; i++)
  {
      if (reverse_string[i] != string[i])
         flag = 0;
  }
   if (flag == 1)
     printf ("%s is a palindrome \n", string);
  else
    printf("%s is not a palindrome \n", string);
}
```

Input:

```
Enter a string
 madam
```

Output:

```
The length of the string 'madam' = 5
madam is a palindrome
```

Record at least 3 results

Signature of faculty with date