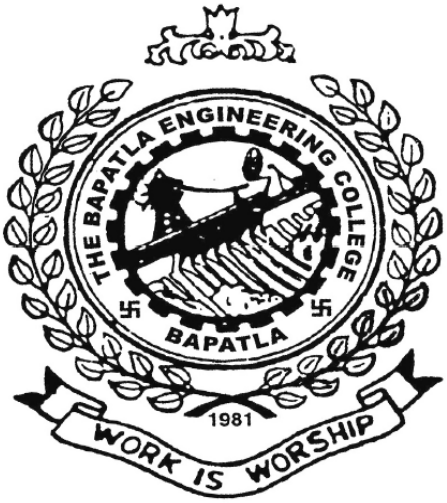
COMPUTER PROGRAMMING WITH C

LAB CODE: EC-153



T Srinivasa Rao
Lecturer, ECE.

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG
BAPTLA ENGINEERING COLLEGE
(AUTONOMOUS)
BAPATLA -522101

LAB COURSE DESCRIPTION

1.Course code: EC153

2. Course Title: Computer Programming Lab with C

3. Core ☐ Elective ☐

4 .pre-requisites (if any): C Language

5. Semester / year in which offered: I year I Semester

6. No. of weeks of Instruction: 39 (sec 'A'), 39 (sec 'B')

7. No. of hours per week: 3

8. Course objective:

9. List of programs: separate sheet has been attached

10. Evaluation procedure

INDEX	DESCRIPTION EVALUATION	TOTAL MARKS
A	Marks allotted for day-to-day lab work .Max : 10 M per 1 program	15 M
B	Marks allotted for record Max : 10 M per 1 program	5 M
C	Marks awarded for viva-voce/quiz	5 M
D	Marks Awarded for Lab Exam	15 M
E	Final sessional marks(A+B+C+D)	40 M
F	University Examination Marks	60 M

INDEX

PROGRAM 1. WRITE A PROGRAM FOR ELECTRICITY BILL TACKING DIFFERENT CATEGORIES OF USERS, DIFFRENT SLABS IN EACH CATEGORY.(USING NESTED IF ELSE STATEMENT

Domestic		Non-Domestic	
Range	Unit per charge	Range	Unit per charge
0 - 200	0.5	0 - 100	0.5
201 - 400	100 + 0.65	101 - 200	50 + 0.60
401 -600	230 + 0.80	201 - 300	100 + 0.70
601 and Above	390 + 1.00	301 and Above	200 + 1.00

```
/*WRITE A PROGRAM FOR ELECTRICITY BILL TACKING DIFFERENT CATEGORIES OF USERS, DIFFRENT SLABS IN EACH CATEGORY.(USING NESTED IF ELSE STATEMENT*/
#include<stdio.h>
main()
{
	int n,cr,pr,mno;
	char utype;
	float bill=0.0;
	printf("\n\t*****************************************");
	printf("\n\t*****ELECTRICITY BILL CALCULATION*******");
	printf("\n\t*****************************************");
	printf("\nEnter meter number:");
	scanf("%d",&mno);

	/*Reading current and previous readings*/
	printf("\nEnter current reading:");
	scanf("%d",&cr);
	printf("\nEnter previous reading:");
	scanf("%d",&pr);
	/*Computing number of units consumed*/
	n=cr-pr;
	printf("\n\t************MENU*****************");
	printf("\n\tEnter D for domestic user");
	printf("\n\tEnter N for non domestic user");
	printf("\n\t*********************************");
	/*Reading type of the user.*/
	printf("\n\nEnter the type of the user:");
	scanf("%s",&utype);

	if(utype=='D'|| utype=='d')
	{
		if(n>=0&&n<=200)
			bill=n*0.50;
		else if(n>200&&n<=400)
			bill=100+n*0.65;
		else if(n>400&&n<=600)
			bill=230+n*0.80;
		else if(n>600)
			bill=390+n*1.0;
		else
			printf("\nInvalid number of units");
```

```
        }

        else if(utype=='N'|| utype=='n')
        {
                if(n>=0&&n<=100)
                        bill=n*0.50;
                else if(n>100&&n<=200)
                        bill=50+n*0.60;
                else if(n>200&&n<=300)
                        bill=100+n*0.70;
                else if(n>300)
                        bill=200+n*1.0;
                else
                        printf("\nInvalid number of units");
        }
        else
        {
                printf("\nInvalid user type");
        }

        if(bill>0)
        {
                printf("\nThe meter number=%d",mno);
                printf("\nThe number of units consumed=%d",n);
                printf("\nBill amount for %d units is %f\n",n,bill);
        }

}/*End of main*/
```

PROGRAM 2. WRITE A PROGRAM TO EVALUATE THE FOLLOWING USING LOOPS

A) 1+X2/2!+X4/4!+...UPTO 5 TERMS

B) X+X3/3!+X5/5!+...UPTO 5 TERMS

```
/*TO EVALUATE  1+X2/2!+X4/4!+...UPTO 5 TERMS*/

#include<stdio.h>
#include<math.h>
main()
{
        int i,j,k=2,fact;
        float x,sum=1.0,p;
        printf("\n****************************************");
        printf("\n*******Sum of Even power series:*******");
        printf("\n****************************************");
        printf("\nEnter the value of x:");
        scanf("%f",&x);
        for(i=1;i<=4;i++)
        {
                p=pow(x,k);
                fact=1;
                for(j=1;j<=k;j++)
                {
                        fact=fact*j;
                }
                sum=sum+(p/fact);
                k=k+2;
        }
        printf("\n\t The sum of given series=%f",sum);
        printf("\n***********End of the Program**********\n");
}
```

```
/*EVALUATING   X+X3/3!+X5/5!+...UPTO 5 TERMS*/

#include<stdio.h>
#include<math.h>
main()
{
        float x,sum=0.0,p;
        int i,j,k=1,fact;
        printf("\n****************************************");
        printf("\n*******Sum of Odd power series:*******");
        printf("\n****************************************");
        printf("\nenter the value of x:");
        scanf("%f",&x);
        for(i=1;i<=5;i++)
        {
                p=pow(x,k);
                fact=1;
                for(j=1;j<=k;j++)
                {
                        fact=fact*j;
                }
                sum=sum+p/fact;
                k=k+2;
        }
        printf("\n\tsum of the series=%f",sum);
        printf("\n***********End of the Program***********\n");
}
```

PROGRAM 3. WRITE A PROGRAM TO CHECK WHETHER THE GIVEN NUMBER IS

A) PRIME OR NOT

B) PERFECT OR ABUNDANT OR DEFICIENT

```
/*PROGRAM TO CHECK WHETHER THE GIVEN NUMBER IS PRIME OR NOT*/
#include<stdio.h>
main()
{
        int i=1,count=0,n;
        printf("\n************************************************");
        printf("\n******To check the given no. is Prime or not*****");
        printf("\n************************************************");
        printf("\nEnter a number:");
        scanf("%d",&n);
        while(i<=n)
        {
                if(n%i==0)
                        count++;
                i++;
        }
        printf("\n****************************************");
        if(count==2)
                printf("\n\tGiven number %d is prime",n);
        else
                printf("\n\tGiven number %d is not a prime",n);
        printf("\n***********End of the Program***********\n");
 }
```

```
/* PROGRAM TO CHECK WHETHER THE GIVEN NUMBER IS PERFECT OR ABUNDANT OR DEFICIENT*/
#include<stdio.h>
main()
{
        int n,sum=0,i=1;
        printf("\n************************************************************************
**");
        printf("\n******To check the given no. is Perfect or Abundent or Deficient*******");
        printf("\n************************************************************************
***");
        printf("\nenter the value of n:");
        scanf("%d",&n);
        for(i=1;i<n;i++)
        {
                if(n%i==0)
                        sum=sum+i;
        }
        printf("\n************************************************");
        if(sum==n)
                printf("\n\tGiven number is perfect number");
        else if(sum>n)
                printf("\n\tGiven number is deficient");
        else
                printf("\n\tGiven number is abundant");
   printf("\n***********End of the Program***********\n");
}
```

PROGRAM 4. WRITE A C PROGRAM TO FIND THE MEAN, MODE, MEDIAN, AND VARIANCE OF LIST OF VALUES BY USING ONE DIMENSIONAL ARRAY

```
/*PROGRAM TO FIND THE MEAN, MODE, MEDIAN, AND VARIANCE OF LIST OF VALUES BY USING ONE DIMENSIONAL ARRAY*/
#include<stdio.h>
#include<math.h>
main()
{
	int i,j,k,a[20],n;
	float mean,med,mode,var=0.0,sum=0.0;
	printf("Enter size of the array:");
	scanf("%d",&n);
	/*reading elemens for array*/
	for(i=0;i<n;i++)
	{
		printf("enter %d th element of the list:",i);
		scanf("%d",&a[i]);
		sum=sum+a[i];
	}
	printf("Elements in the array are:\n");
	for(i=0;i<n;i++)
		printf("%d\t",a[i]);
	/* here we are calculating mean*/
	mean=sum/n;
	/* here we are sorting the elements in the array*/
	for(i=0;i<n;i++)
		for(j=i+1;j<n;j++)
			if(a[i]>a[j])
			{
				k=a[i];
				a[i]=a[j];
				a[j]=k;
			}
	printf("\nSorted elements are:\n");
	for(i=0;i<n;i++)
		printf("%d\t",a[i]);
	/*here we are calculating median*/
	if(n%2==0)
		med=(a[n/2]+a[(n/2)-1])/2;
	else
		med=a[(n-1)/2];
	/*calculating mode*/
	mode=3*med-2*mean;
	/*here we are calculating variance*/
	for(i=0;i<n;i++)
		var=var+pow((mean-a[i]),2);
	printf("\nmean=%f,med=%f,mode=%f,var=%f\n",mean,med,mode,var/n);
}
```

PROGRAM 5. WRITE A C PROGRAM TO READ A LIST OF NUMBERS AND PERFORM THE FOLLOWING OPERATIONS

A) PRINT THE LIST
B) DELETE DUPLICATES FROM THE LIST
C) REVERSE THE LIST

```
/*PROGRAM TO READ A LIST OF NUMBERS AND PERFORM THE FOLLOWING OPERATIONS
        A) PRINT THE LIST          B) DELETE DUPLICATES FROM THE LIST   C) REVERSE THE LIST*/
#include<stdio.h>
#include<stdlib.h>

/*function prototype declaration*/
void printmenu();
void printlist(int a[],int b);
void printreverse(int a[],int b);
int deletedup(int a[],int b);

main()
{
        int a[20],n;
        int i,j,ele,l,temp;
        int choice;
        printf("Enter the size of the array(no. of elements to be stored):");
        scanf("%d",&n);
        printf("Enter your array elements\n");
        for(i=0;i<n;i++)
        {
                printf("Enter %d th element:",i);
                scanf("%d",&a[i]);
        }

        do{
        /*To print the menu*/
        printmenu();
        printf("Enter your choice:");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1:
                        printlist(a,n);
                        break;
                case 2:
                        n=deletedup(a,n);
                        printlist(a,n);
                        break;
                case 3:
                        printreverse(a,n);
                        break;
                case 4:
                        exit(0);
                        break;
```

```
                default:
                        printf("\n Invalid option ");
                        break;
        } /*End for switch*/
        }while(choice!=4);

} /*End for main*/

void printmenu()
{
        printf("\n**********USER MENU**********\n");
        printf("Enter 1 for Print the List\n");
        printf("Enter 2 for Delete duplicates from the list\n");
        printf("Enter 3 for Reverse the List\n");
        printf("Enter 4 for Exit from the program\n");
        printf("*****************************************\n");
}
void printlist(int a[],int n)
{
        int i;
        printf("\nArray elements are:\n");
        for(i=0;i<n;i++)
                printf("%d\t",a[i]);
}
void printreverse(int a[],int n)
{
        int i;
        printlist(a,n);
        printf("\nArray elements in reverse order are:\n");
        for(i=n-1;i>=0;i--)
                printf("%d\t",a[i]);
}
int deletedup(int a[],int n)
{       int i,j,flag=0,k;
        for(i=0;i<n;i++)
                for(j=i+1;j<n;j++)
                        if(a[i]==a[j])
                        {       flag=1;
                                for(k=j;k<n;k++)
                                        a[k]=a[k+1];
                                j--;
                                n--;
                        }
        if(flag!=1)
        {       printf("no duplicates");
                return n;
        }
        else
                return n;
}
```

PROGRAM 6. WRITE A PROGRAM TO READ A LIST OF NUMBERS AND SEARCH FOR GIVEN NUMBER USING BINARY SEARCH ALGORITHM AND IF FOUND DISPLAY ITS INDEX OTHERWISE DISPLAY THE MESSAGE "ELEMENT NOT FOUND IN THE LIST"

/*PROGRAM TO READ A LIST OF NUMBERS AND SEARCH FOR GIVEN NUMBER USING BINARY SEARCH ALGORITHM*/

```
#include<stdio.h>

int binarysearch(int a[], int b, int c);
void sortelements(int a[],int c);

main()
{
	int a[20],n;
	int i,j,ele,l,temp;
	printf("Enter the size of the array(no. of elements to be stored):");
	scanf("%d",&n);
	printf("Enter your array elements\n");
	for(i=0;i<n;i++)
	{
		printf("Enter %d th element:",i);
		scanf("%d",&a[i]);
	}
	/*Calling function to sort the elementsin the list*/
	sortelements(a,n);

	do{
		printf("\nEnter an element to search:");
		scanf("%d",&ele);
		/*Searching an element by using binary search algorithm*/
		l=binarysearch(a,n,ele);
		if(l==0)
			printf("\nGiven element %d is not found\n",ele);
		else
			printf("\nGiven element %d is found at %d th position\n",ele,l);
	}while(ele!=-1);
} /*End of main*/

void sortelements(int a[],int n)
{
	int i,j,temp;
	for(i=0;i<n;i++)
		for(j=i;j<n;j++)
			if(a[i]>a[j])
			{
				temp=a[i];
				a[i]=a[j];
				a[j]=temp;
			}
```

```
	printf("\nArray element after sorting:\n");
	for(i=0;i<n;i++)
		printf("%d\t",a[i]);
}

int binarysearch(int a[],int n,int e)
{
	int i,low=0,high,mid;
	high=n-1;
	do{
		mid=(low+high)/2;
		if(a[mid]>e)
			high=mid;
		else if(a[mid]<e)
		{
			mid++;
			low=mid;
		}
	}while(a[mid]!=e && low<high);
	if(a[mid]==e)
		return mid+1;
	else
		return 0;
}
```

PROGRAM 7. WRITE A MENU DRIVEN PROGRAM TO READ TWO MATRICES AND COMPUTE THEIR SUM AND PRODUCT USING FUNCTIONS

/*PROGRAM TO READ TWO MATRICES AND COMPUTE THEIR SUM AND PRODUCT USING FUNCTIONS*/

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

/*Function prototype here first dimension is optional */
void printmenu();
void readmatrix(int m[][10],int b,int c);
void printmatrix(int m[][10],int b,int c);
void findsum(int a[][10],int b[][10],int c,int d);
void findproduct(int a[][10],int b[][10],int c,int d,int e);
main()
{
	int a[10][10],b[10][10];
	int r1,c1,r2,c2,choice;
	/*reading first matrix*/
	printf("Enter size of the matrix a(rows and columns):");
	scanf("%d%d",&r1,&c1);
	readmatrix(a,r1,c1);
	/*reading second matrix*/
	printf("Enter size of the matrix b(rows and columns):");
	scanf("%d%d",&r2,&c2);
	readmatrix(b,r2,c2);
	do{
		printf("\n");
		/*to print menu*/
		printmenu();
		printf("enter your choice:");
		scanf("%d",&choice);
		switch(choice)
		{
			case 1: /*printing matrices*/
				printmatrix(a,r1,c1);
				printmatrix(b,r2,c2);
				break;
			case 2:
				if(r1==r2 && c1==c2)
					findsum(a,b,r1,c1);
				else
					printf("Addition is not possible\n");
				break;
			case 3:
				if(c1==r2)
					findproduct(a,b,r1,c1,c2);
				else
					printf("Multiplication is not possible\n");
				break;
```

```
                        case 4:
                                printf("\t***End of the program***\n");
                                exit(0);
                                break;
                        default:
                                printf("Invalid user option");
                                break;
                }/* closing for switch*/

        }while(choice!=4);

}/*ending for main function*/

/*to print the menu*/
void printmenu()
{
        printf("\t****************** USER MENU ****************\n");
        printf("\tEnter 1 for print the two matrices\n");
        printf("\tEnter 2 for print the sum of two matrices\n");
        printf("\tEnter 3 for print the product of two matrices\n");
        printf("\tEnter 4 for exit\n");
        printf("\t*********************************************\n");
}

/*to read matrix*/
void readmatrix(int m[][10],int r,int c)
{
        int i,j;
        for(i=0;i<r;i++)
                for(j=0;j<c;j++)
                {
                        printf("enter %d th row and %d th column element:",i,j);
                        scanf("%d",&m[i][j]);
                }
}

/*to print matrix*/
void printmatrix(int m[][10],int r,int c)
{
        int i,j;
        printf("The elements in the matrix are:\n");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        printf("%d\t",m[i][j]);
                }
        printf("\n");
        }
}
```

```
/*to find product of two matrices*/
void findproduct(int a[][10],int b[][10],int r,int d,int c)
{
	int i,j,k;
	int m[10][10];
	for(i=0;i<r;i++)
		for(j=0;j<c;j++)
		{
			m[i][j]=0;
			for(k=0;k<d;k++)
				m[i][j]=m[i][j]+a[i][k]*b[k][j];
		}
	printf("\t**********************************\n");
	printf("\tThe product of two matrixes is:\n");
	printf("\t**********************************\n");
	printmatrix(m,r,c);
}

/*to find sum of two matrices*/
void findsum(int a[][10],int b[][10],int r,int c)
{
	int i,j;
	int d[10][10];
	for(i=0;i<r;i++)
		for(j=0;j<c;j++)
			d[i][j]=a[i][j]+b[i][j];
	printf("\t**********************************\n");
	printf("\tThe sum of two matrixes is:\n");
	printf("\t**********************************\n");
	printmatrix(d,r,c);
}
```

PROGRAM 8. WRITE A MENU DRIVEN PROGRAM TO READ LIST OF STUDENT NAMES AND PERFORM THE FOLLOWING OPERATIONS USING FUNCTIONS.

A) TO PRINT LIST OF NAMES
B) TO SORT THEM IN ASCENDING ORDER
C) TO PRINT THE LIST AFTER SORTING

```
/* PROGRAM TO READ LIST OF STUDENT NAMES AND PERFORM THE FOLLOWING OPERATIONS USING FUNCTIONS.  A) TO PRINT LIST OF NAMES B) TO SORT THEM IN ASCENDING ORDER C) TO PRINT THE LIST AFTER SORTING */
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>

/*function prototypes*/
void printmenu();
void printlist(char str[][20],int n);
void sortlist(char str[][20],int n);

void main()
{
	char str[20][20];
	int i,j,n,choice;
	char permision;
	int sorted;
	/* to read the strings*/
	printf("\t Enter the number of strings to be read:");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("\t Enter %d th student name:",i);
		scanf("%s",str[i]);
	}
	do{
	/*to print the menu */
	printmenu();
	printf("\t Enter your option: ");
	scanf("%d",&choice);
	switch(choice)
	{
		case 1:
			printlist(str,n);
			break;
		case 2:
			sortlist(str,n);
			sorted=1;
			printf("\n*****************************************\n");
			printf("The given list is successfully sorted\n");
			printf("Choose option 3 to print the sorted list\n");
			printf("*****************************************\n");
```

```
                break;
        case 3:
                if(sorted==1)
                {
                        printf("The sorted list elements are:\n");
                        printlist(str,n);
                }
                else
                        printf("\nSorting is not performed\n");

                break;
        case 4:
                exit(0);
                break;
        default:
                printf("Invalid user option");
                break;
    }/*end of switch*/

                printf("\tDo you want to continue(y/n):");
                scanf("%s",&permision);
    }while(permision=='y');

}/*end of main*/

void printmenu()
{
    /*to print the menu */
    printf("\n\t***********************************************\n");
    printf("\t********Choose any one of the following*******\n");
    printf("\t***********************************************\n");
    printf("\t Enter 1 To print the list of names\n ");
    printf("\t Enter 2 To sort the names in alphabetical order\n");
    printf("\t Enter 3 To print list of names after sorting\n");
    printf("\t Enter 4 To exit.\n");
    printf("\t***********************************************\n");
}

/*to print the contents of the list*/
void printlist(char str[][20],int n)
{
    int i;
    printf("\t*******************************\n");
    printf("\tNo.\t\t\tName");
    printf("\n\t*******************************\n");
    for(i=0;i<n;i++)
    {
        printf("\t%d \t\t\t%s\n",i+1,str[i]);
    }
}
```

```
/*to sort the list*/
void sortlist(char str[][20],int n)
{
	int i,j;
	char name[20];
	for(i=0;i<n;i++)
		for(j=i+1;j<n;j++)
			if(strcmp(str[i],str[j])>0)
			{
				strcpy(name,str[i]);
				strcpy(str[i],str[j]);
				strcpy(str[j],name);
			}

}
```

PROGRAM 9. WRITE A C PROGRAM THAT CONSISTS OF RECURSIVE FUNCTIONS TO FIND

A) FACTORIAL OF A GIVEN NUMBER
B) PRINT THE PASCAL TRIANGLE USING BIONOMIAL THEOREM

/*A) PROGRAM TO FIND FACTORIAL OF A GIVEN NUMBER*/

```
#include<stdio.h>
long findfact(int a);
main()
{
        int n;
        long fact;
        do{
        printf("\nEnter a number to find factorial:");
        scanf("%d",&n);
        /*function call to find factorial*/
        fact=findfact(n);
        printf("\n Factorial of the given number %d is %ld\n",n,fact);
        }while(n>=0);
}

/*to find factorial*/
long findfact(int n)
{
        if(n>0)
                return n*findfact(n-1);
        else
                return 1;
}
```

```
/*B) PROGRAM TO PRINT THE PASCAL TRIANGLE USING BIONOMIAL THEOREM */
#include<stdio.h>
int bioncoe(int a,int b);
main()
{
	int n,m,i,j;
	printf("\nEnter a number to print pascal triangle:");
	scanf("%d",&n);
	m=2*n;
	for(i=0;i<=n;i++)
	{
		for(j=1;j<=m;j++)
			printf(" ");
		for(j=0;j<=i;j++)
			printf("%4d", bioncoe(i,j));
		printf("\n");
		m=m-2;

	}
}

int bioncoe(int n,int r)
{
	if(r==0)
		return 1;
	else if(n==r)
		return 1;
	else
		return bioncoe(n-1,r)+bioncoe(n-1,r-1);
}
```

PROGRAM 10. WRITE A MENU DRIVEN PROGRAM TO READ LIST OF STUDENT NAMES AND PERFORM THE FOLLOWING OPERATIONS USING ARRAY OF CHARECTER POINTERS.

A) TO INSERT A STUDENT NAME
B) TO DELETE A NAME
C) TO PRINT THE NAMES

```
/* PROGRAM TO READ LIST OF STUDENT NAMES AND PERFORM THE FOLLOWING OPERATIONS USING
ARRAY OF CHARECTER POINTERS.  A) TO INSERT A STUDENT NAME  B) TO DELETE A NAME  C) TO
PRINT THE NAMES */

#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>

/*function prototypes*/
void printmenu();
void printlist(char *s[],int *a);
void insertname(char *s[],int *a);
void deletename(char *s[],int *a);
void main()
{
        char *str[20],name[20];
        int *n,m,length,i,choice;
        printf("enter how many strings should be read:");
        scanf("%d",&m);

        //assigning size into a pointer
        n=&m;
        //Reading names into list.
        for(i=0;i<*n;i++)
        {
                printf("Enter %d th name:",i);
                scanf("%s",name);
                // finding length of the string
                length=strlen(name);
                //Assigning memory to store the string
                *(str+i)=(char*)malloc(length+1);
                //coping string to the assigned block
                strcpy(*(str+i),name);
        }

        do{
                //to print the user menu
                printmenu();
                printf("Enter your choice:");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
```

```
                        printf("\nThe names in the list are:\n");
                        printlist(str,n);
                        break;
                case 2:
                        insertname(str,n);
                        printf("\nList after inserting a name is:\n");
                        printlist(str,n);
                        break;
                case 3:
                        deletename(str,n);
                        break;
                case 4:
                        break;
                default: printf("\nInvalid user option");
                        break;
            }

        }while(choice!=4);

        printf("\n**********End of the Program**********\n");

}/*End of main*/

void printmenu()
{
        /*to print the menu */
        printf("\n\t************************************************\n");
        printf("\t********Choose any one of the following********\n");
        printf("\t************************************************\n");
        printf("\t Enter 1 To Print the list of names\n ");
        printf("\t Enter 2 To Insert a student name into list\n");
        printf("\t Enter 3 To Delete a student name from the list\n");
        printf("\t Enter 4 To Exit.\n");
        printf("\t************************************************\n");
}

//Function to print the list
void printlist(char *s[],int *a)
{
        int i;
        printf("\t*******************************\n");
        printf("\tNo.\t\t\tName");
        printf("\n\t*******************************\n");
        for(i=0;i<*a;i++)
        {
                printf("\t%d \t\t\t%s\n",i+1,*(s+i));
        }
}
// Function to insert a name into list
void insertname(char *s[],int *a)
```

```
{
        int pos,i;
        char name[20];
        printf("Enter a name to be inserted:");
        scanf("%s",name);
        printf("Enter a position where the name will be inserted:");
        scanf("%d",&pos);
        /*moving strings*/
        for(i=*a-1;i>=pos-1;i--)
                strcpy(*(s+i+1),*(s+i));
        strcpy(*(s+pos-1),name);
        //incrementing the size of the list
        (*a)++;
        printf("\nGiven name is inserted successfuly");
}

//Function to delete specified name from the list
void deletename(char *s[],int *a)
{
        int pos,i,j,flag=0;
        char name[20];
        printf("Enter a name to be deleted:");
        scanf("%s",name);
        for(i=0;i<*a;i++)
        {
                if(strcmp(name,*(s+i))==0)
                {
                        flag=1;
                        for(j=i;j<*a-1;j++)
                                strcpy(*(s+j),*(s+j+1));
                        (*a)--;  //decreasing size of the list
                }
        }
        if(flag==1)
        {
                printf("\nGiven name is deleted successfuly");
                printf("\nList after deleting a name is:\n");
                printlist(s,a);
        }
        else
                printf("\nGiven name is not found in the list\n.");

}
```

*******All the Best*******