Home Assignments:

Day 7:

Q1: Write a program to find the non-repeated element in an array. The array contains n number of values where all the elements (except one) appear exactly twice.

Soln:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
int n;
printf("Enter the value of n : \n");
scanf("%d",&n);
int a[n];
printf("Enter the elements in the array : \n");
for(int i=0;i<n;i++)
   {
scanf("%d",&a[i]);
   }
printf("The elements in the array are : \n");
for(int i=0;i<n;i++)
   {
printf("%d ",a[i]);
   }
int p=a[0];
for(int i=1;i<n;i++)
   {
p^=a[i];
   }
printf("\nThe element in the array which appears only once is : \n");
printf("%d\n",p);
return 0;
}
```

Q2: Write a program to print a histogram of the frequencies of different characters present in a character array entered through keyboard.

Soln:

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

```
#include <string.h>

void histogram(char arr[]){
int i, j, count =0;
for(i=0; i<26;i++){
for(j=0; arr[j]!='\0'; j++){
if(arr[j]==65+i || arr[j]==97+i)
count++;
        }
if(count!=0)
printf("The character %c is repeated in the array : %d times\n", 65+i, count);
count = 0;
    }

}
int main(){
int n,i,j, count;

printf("\nEnter the number of elements : ");
scanf("%d",&n);
char arr[n];
printf("\nInput the array elements : \n");
fflush(stdin);
gets(arr);
histogram(arr);
return 0;
}
```

Q3:Write a program to find the largest and second largest from a set of numbers.

Soln:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
int n;
printf("Enter the value of n : \n");
scanf("%d",&n);
int a[n];
printf("Enter the elements in the array : \n");
for(int i=0;i<n;i++)
    {
scanf("%d",&a[i]);
    }
printf("The elements in the array are : \n");
for(int i=0;i<n;i++)
    {
printf("%d ",a[i]);
    }
printf("\n");
int largest=a[0];
```

```
int j=0;
for(int i=1;i<n;i++)
   {
if(a[i]>largest)
       {
largest=a[i];
           j=i;
       }
   }
int secondlargest=a[0];
for(int i=1;i<n;i++)
    {
if(i==j)
       {
continue;
       }
else if(a[i]>secondlargest)
       {
secondlargest=a[i];
       }
    }
printf("Largest element in the array is : \n");
printf("%d\n",largest);
printf("Second Largest element in the array is : \n");
printf("%d\n",secondlargest);
return 0;
}
```

Day 8:

Q1; Write a program to find whether a matrix is orthogonal or not.

Soln:

```
#include<stdio.h>
int Orthogonal(int n, int a[20][20])
{
int prod[n][n], i, j, k, sum;
for ( i = 0; i < n; i++)
   {
for ( j = 0; j < n; j++)
           {
sum = 0;
for (int k = 0; k < n; k++)
           {
sum += (a[i][k] * a[j][k]);
```

```
            }
if (i != j && sum != 0)
return 0;
if (i == j && sum != 1)
return 0;
        }
    }
return 1;
}
int main()
{
int rows, column, i, j;
printf("Enter the rows of the matrix : ");
scanf("%d", &rows);
printf("Enter the column of the matrix : ");
scanf("%d", &column);
printf("Enter the elements of the matrix \n");
int a[20][20];
for(i =0; i<rows; i++){
printf("Enter the elements for row %d\n", i+1);
for(j=0; j<column; j++)
scanf("%d", &a[i][j]);
    }
if(rows != column)
printf("The matrix is not orthogonal\n");
else
    {
if(Orthogonal(rows, a))
        {
printf("The matrix is orthogonal\n");
        }
else
        {
printf("The matrix is not orthogonal\n");
        }
    }

}
```

Q2: Write a program to find whether a given matrix is upper triangular or lower triangular.

Soln:

```
#include<stdio.h>
int isLowerT(int X[][10],int n)
{
int i,j,f=1;
for(i=0;i<n;i++)
    {
for(j=i+1;j<n;j++)
if(X[i][j]!=0)
            {
                f=0;
```

```
        }
if(f==0)
break;
   }
return f;
}

int isUpperT(int X[][10],int n)
{
int i,j,f=1;
for(i=0;i<n;i++)
   {
for(j=0;j<i;j++)
if(X[i][j]!=0)
        {
            f=0;
        }
if(f==0)
break;
   }
return f;
}
int main()
{
int n,A[10][10],i,j;
printf("Enter the number of rows or columns of the square Matrix : ");
scanf("%d",&n);
printf("Enter the elements of the matrix : \n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
scanf("%d",&A[i][j]);
printf(isUpperT(A,n)?"Matrix is Upper Triangular and ":"Matrix is not Upper Triangular and ");
printf(isLowerT(A,n)?"\nMatrix is Lower Triangular ":"\nMatrix is not Lower Triangular");

}
```

Day 9:

Q1) Write a program to read a line of text from the keyboard and convert it into a coded text by changing its characters by adding a code number to them. This code number must be taken as input

Soln:

```
#include<stdio.h>
char* encode(char *p,int n);
int main()
{
        int n;
        char p1[100];
        printf("enter the string\n");
        gets(p1);
        char *p=p1;
        printf("enter the no. to be added to the string:");
```

```
	scanf("%d",&n);
	encode(p,n);
	printf("%s",p);
	return 0;
}
char* encode(char *p,int n)
{
	while(*p!='\0')
	{
		*p=*p+n;
		p++;
	}
	return p;
}
```

Q2) Write a program to read in a line of text and count the number of blank spaces, tabs and new lines in that line. Also rewrite the line of text with tabs and new lines replaced by the visible sequences ‘\t’ ‘\n’

Soln:

```
#include<stdio.h>
char* count(char *p);
int main()
{
	char p1[100];
	int *a;
	printf("enter the string\n");
	gets(p1);
	char *p=p1;
	count(p);
	printf("%s",p);
	return 0;
}
char* count(char *p)
{
	int a[3];
	a[0]=0;a[1]=0;a[2]=0;
	while(*p !='\0')
	{
		if(*p=='\t')
		{
			a[0]++;
			*p='T';
		}
		if(*p==' ')
		{
			a[1]++;
		}
		if(*p=='\n')
		{
			a[2]++;
			*p='N';
		}
		p++;
```

```
        }
        printf("tabs=%d,space=%d,new line=%d\n",a[0],a[1],a[2]);
        return p;
}
```

Q3) Write a program to read in a line of text and count the number of lines, words and characters in that text.

Soln:

```
#include<stdio.h>
int count(char *p);
int main()
{
        char p1[100];
        printf("enter the string\n");
        gets(p1);
        char *p=p1;
        count(p);
        return 0;
}
int count(char *p)
{
        int a[3];
        a[0]=0;a[1]=0;a[2]=0;
        while(*p !='\0')
        {
                if((*p<65)||(*p>90&&*p<97)||(*p>122))
                        a[0]++;
                if((*p=='.')||(*p=='?')||(*p=='!'))
                        a[2]++;
                p++;
                a[1]++;
        }
        printf("words=%d,characters=%d,lines=%d\n",a[0],a[1],a[2]);
}
```

Q4) Write a program to count the number of vowels and digits in a given string.

Soln:

```
#include<stdio.h>
int count(char *p);
int main()
{
        char p1[100];
        printf("enter the string\n");
        gets(p1);
        char *p=p1;
        count(p);
        return 0;
}
int count(char *p)
{
```

```
        int a[2];
        a[0]=0;a[1]=0;
        while(*p !='\0')
        {
                if((*p=='a')||(*p=='e')||(*p=='i')||(*p=='o')||(*p=='u'))
                {
                        a[0]++;
                }
                if((*p>=48)&&(*p<=57))
                        a[1]++;
                p++;
        }
        printf("vowels=%d,digits=%d\n",a[0],a[1]);
}
```

Day 10:

Q1: Write a recursive function to evaluate n Cr

Soln:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int fact(int n)
{
if(n==0)
  {
return 1;
  }
return n*fact(n-1);
}

int main()
{
int n;
printf("Enter the value of n : \n");
scanf("%d",&n);
int r;
printf("Enter the value of r : \n");
scanf("%d",&r);
if(n>=r)
   {
int k=fact(n)/(fact(n-r)*fact(r));
printf("The value of nCr is : %d\n",k);
   }
```

```
else
   {
printf("Invalid input!\n");
   }

}
```

Q2: Write a recursive function to convert a decimal integer, taken as input, to its hexadecimal equivalent.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
int Hexa(int n,int i)
{
int m=0;
if(i<0)
return 0;
else
   {
   m=n>>(4*i);
if(m>=0 && m<=9)
printf("%d",m);
else if(m==10)
printf("A");
else if(m==11)
printf("B");
else if(m==12)
printf("C");
else if(m==13)
printf("D");
else if(m==14)
printf("E");
else if(m==15)
printf("F");
return Hexa((n%(1<<4*i)),i-1);
   }
}
int main()
{
int n,b,i=3;
printf("Enter number  in decimal :  ");
scanf("%d",&n);
printf("\nNumber. in HexaDecimal : 0x");
   b=Hexa(n,i);
return 0;
}
```

Q3: Find the maximum and minimum of a list of numbers using a recursive function.
Soln:

```
int max(int n, int arr[n], int maxi, int i){
```

```
if(i==n)
return maxi;
else{
maxi = arr[i]>maxi?arr[i]:maxi;
return max(n, arr, maxi, i+1);
    }
}

int min(int n, int arr[n], int mini, int i){
if(i==n)
return mini;
else{
mini = arr[i]<mini?arr[i]:mini;
return min(n, arr, mini, i+1);
    }
}

int main(){
int i, n;
printf("Please enter the size of the array : ");
scanf("%d", &n);
int a[n];
printf("Please enter the elements of the array : \n");
for(i =0; i<n; i++)
    {
scanf("%d", &a[i]);
    }
printf("The maximum element is : %d\n", max(n, a, a[0], 1));
printf("The minimum element is : %d\n", min(n, a, a[0], 1));
}
```

Q4: Write a recursive function to calculate the sum of all digits of a number entered by the user.

Soln:

```
#include<stdio.h>
int sum(int n)
{
if(n==0)
return 0;
return (n%10 + sum(n/10));
}
int main()
{
int n;
printf("Enter the number :");
scanf("%d",&n);
printf("Sum of digits = %d \n",sum(n));
return 0;
}
```

Day 11:

Q 1) Write an interactive C program, using pointers, which will encode a line of text. To encode a line of text, proceed as follows: Convert each character, including blank spaces, to its ASCII equivalent Generate a positive random integer. Add this integer to the ASCII equivalent of each character. The same random integer will be used for the entire line of text

 Suppose that N1 represents the lowest permissible value in the ASCII code, and N2 represents the highest permissible value. If the number obtained in step 2 above (i.e., the original ASCII equivalent plus the random integer) exceeds N2, then subtract the largest possible multiple of N2 from this number, and add the remainder to N1. Hence the encoded number will always fall between N1 and N2, and will therefore always represent some ASCII character. Display the characters that correspond to the encoded ASCII values.

Soln:

```
#include <stdio.h>

#include <string.h>

#include <stdlib.h>



char* encode(char* ptr, int len);



int main()

{

        int len;

        char str[50];


        printf("Enter the line of text \n");

        scanf("%[^\n]%*c",str);


        len = strlen(str);
```

```
	encode(str,len);
	puts(str);

	return 0;
}


char* encode(char* ptr,int len)
{
	int p, i=0, x=20;

	srand(2);
	x=rand();

	while(i<len)
	{
		p=*(ptr+i)+x;
		if(p>255)
		{
			p=p%224+32;
		}
	}

	*(ptr+i)=p;
	i++;

	return ptr;
```

```
}
```

Q2) Write a program, using pointers, that accepts a string and converts all its characters to upper or lower case. Use the functions toupper( ) and tolower( ) defined in ctype.h

Soln:

```
#include <stdio.h>

#include <ctype.h>

#include <stdlib.h>


int main()

{

	char *p=(char *)calloc(100,sizeof(char));

	int i;

	char *t;

	char te;

	t=p;

	printf("Start entering the sentence:\n");

	for(i=0; i<100; i++)

	{

		scanf("%[^\n]",p);

	}

	p=t;

	for(i=0; i<100; i++)

	{

		te=*(p++);

		*p=toupper(te);

	}
```

```
	p=t;
	for(i=0; i<100; i++)
		printf("%c",*(p++));
	free(p);
	return 0;
}
```

Q3) Write three functions, using pointers, to concatenate two strings, to compare two strings and to reverse a string, respectively. Test these functions in a complete program.

Soln:

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

int compr(char* a,char* b);
char* concate(char* a,char* b);
int revrse(char* a,int len);

int main()
{
	int choice,a,z;
	char* p;
	char str1[50],str2[50];
```

```
printf("Enter Your Choice for:\n 1.Comparison of two strings\n");
printf("2.Concatenation\n 3.Reversal of the string\n");
scanf("%d",&choice);


switch(choice)
{
        case 1:

                printf("Enter the 1st string : \n");
                scanf("%s",str1);
                printf("Enter the 2nd string : \n");
                scanf("%s",str2);

                a=compr(str1,str2);

                if(a==0)
                printf("Same string\n");

                else
                printf("Different string\n");
        break;

        case 2:

                printf("Enter the first string : \n");
```

```
        scanf("%s",str1);
        printf("Enter the second string : \n");scanf("%s",str2);
        printf("The concatenated string is :\n");

        p=concate(str1,str2);

        if(p==str2)
        printf("Not enough memory");

        else
        printf("%s",p);
break;

case 3:

        printf("Enter the string : \n");
        scanf("%s", str1);

        while(str1[z]!='\0')
        {
        z++;
        }

        revrse(str1,z);
        puts(str1);
break;
```

```
		default:

			printf("Wrong choice");


		return 0;
	}
}

int compr(char* a, char* b)
{
	int i,count = 0;


	for(i = 0; ; i++)
	{
		if(a[i]=='\0'||b[i]=='\0')
			if(a[i]!='\0'||b[i]!='\0')
				count++;
		break;
	}


	if(a[i]!=b[i])
		count++;

	return count;
```

```
}

char* concate(char* a, char* b)
{


	int i=0,j=0;


	while(a[i]!='\0')
	{
		i++;
	}

	while(b[j]!='\0')
	{
		a[i+j] = b[j];
		j++;
	}


	a[i+j]='\0';


	if((i+j)>50 )
		return b;
```

```
        else
                return a;
}


int revrse(char* a, int len)
{
        int i,j,b[len];

        for(i=0,j=len-1;j>=0;i++,j--)
        {
                a[i]=b[j];
        }

        for(i=0;i<len;i++)
                a[i]=b[i];

        return 0;
}
```

Day 12:

Q 1) Write a program by using structure that describes the set of books in a library. For each book the members are name of author, publisher, price and branch information. The program should, print the list of books supplied by a publisher print the list of books in a particular branch in a file say “lib.txt”

Soln:

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


struct books
{
	char name[50];
	char publ[25];
	float price;
	char branch[40];
};


int main()
{
	int i = 0, n;
	FILE *fin;
	struct books b[i];
```

```
char add[50];

printf("Enter the name of the file containing th list of books\n");
scanf("%[^\n]%*c", add);

fin = fopen(add, "r");

if (fin == NULL)
{
        fprintf(stderr, "\nError opening the file");
        exit(1);
}
while(fread(&b[i], sizeof(struct books), 1, fin))
{
        i++;
}

n = i;
fclose(fin);

for(i = 0; i < n; i++)
{
        printf("%s ", b[i].name);
        printf("%s ", b[i].publ);
```

```
            printf("%f ", b[i].price);
            printf("%s ", b[i].branch);
            printf("\n");
        }

        FILE *fout;

        char str[20];

        fout = fopen("\\Desktop\\Projects\\HomeAss\\Soln\\12.1\\O_P.txt",
"w");
        printf("Enter the branch name you want to print\n");
        scanf("%[^\n]%*c", str);

        for(i = 0; i < n; i++)
        {
            if(strcmp(b[i].name, str) == 0)
            {
                fwrite(&b[i], sizeof(struct books), 1, fout);
            }
        }

        fclose(fout);

        return 0;
```

```
}
```

Q2) Develop a simple telephone directory which saves your friends contact information in a file named directory.txt.

Soln:

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


typedef struct
{
        char name[40];
        char ph[13];
        char email[50];
}contact;

void towrite();
void toread();

int main()
{
        int choice;
        char n;
```

```
	while(scanf("%c",&n))
	{
		printf("\nEnter:\n1.\tTo write.\n2.\tTo display every contact.\n");
		scanf("%d",&choice);
		switch(choice)
		{

			case 1:
			towrite();
			break;

			case 2:
			toread();
			break;

			default:
			printf("\nERROR:\tWrong INPUT!");
		}
		printf("\nEnter any number to stop \n\t\tOr else enter any character:\n");
	}
	return 0;
}
```

```
void towrite()
{
        FILE *outfile;
        char yn;

   outfile=fopen ("directory.txt", "a");
   if (outfile == NULL)
   {
      fprintf(stderr, "\nERROR: File not opened.\n");
      exit (1);
   }


       do
       {
              contact input;

              printf("\nEnter the name of the person\n");
              scanf("%[^\n]%*c",input.name);
              printf("\nEnter the phone of the person\n");
              scanf("%[^\n]%*c",input.ph);
              printf("\nEnter the e-mail of the person\n");
              scanf("%[^\n]%*c",input.email);
```

```
            fwrite (&input, sizeof(contact), 1, outfile);

            if(fwrite != 0)
                printf("\nContents to file written SUCCESSFULLY!\n");
            else
                printf("\nERROR: writing in file!\n");

            printf("\nDo you want to continue?[y/n]\t");
            scanf("%c",&yn);
        }while(yn=='Y'||yn=='y');

    fclose (outfile);

}

void toread()
{
        FILE *infile;
    contact input;


    infile = fopen ("directory.txt", "r");
    if (infile == NULL)
```

```
    {
        fprintf(stderr, "\nERROR: File not opened.\n");
        exit (1);
    }

    while(fread(&input, sizeof(contact), 1, infile))
        printf ("%s\t%s\t%s\n", input.name,input.ph, input.email);

    fclose (infile);
}
```