

Lab Question Solution for the Degree of Bachelors in Information
Technology

Lab Question Solution



NITIK KUMAR ADHIKARI
(7093)

KIST College
Faculty of Science and Technology
Purbanchal University, Nepal

April, 2024

Pattern:

1.

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

```
int main()
```

```
{
```

```
    int i=0,j=0;
```

```
    char a[]="NEPAL";
```

```
    for(i=0;i<5;i++)
```

```
    {
```

```
        for (j=0;j<=i;j++)
```

```
        {
```

```
            printf("%c",a[j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

N

NE

NEP

NEPA

NEPAL

2.

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

```
int main()
```

```
{
```

```
    int i=0,j=0;
```

```
    for(i=1;i<=5;i++)
```

```
    {
```

```
        for (j=5;j>=i;j--)
```

```
        {
```

```
            printf("%d",j);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

5 4 3 2 1

5 4 3 2

5 4 3

5 4

5

3.

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

```
int main()
```

```
{
```

```
    int k=0,i=0,j=0;
```

```
    for(i=1;i<=5;i++)
```

```
    {
```

```
        for(k=i;k<=4;k++)
```

```
        {
```

```
            printf(" ");
```

```
        }
```

```
        for(j=1;j<=2*i-1;j++)
```

```
        {
```

```
            printf("* ");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

*

* * *

* * * * *

* * * * * *

* * * * * * *

4.

```
#include <stdio.h>
int main()
{
    int i=0,j=0,k=0;
    for (i=5;i>=1;i--)
    {
        for (k=i;k<5;k++)
        {
            printf(" ");
        }
        for (j=1;j<=2*i-1;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

```
* * * * *
  * * * * *
    * * * *
      * * *
        *
```

5.

```
#include <stdio.h>
int main()
{
    int i=0,j=0;
    for(i=1;i<=4;i++)
    {
        for (j=1;j<=4;j++)
        {
            if((i+j)%2==0)
            {
                printf("$ ");
            }
            else
                printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

```
$ * $ *
* $ * $
$ * $ *
* $ * $
```

Structure

1. Write a program with structure "Students" to contain name, faculty, email, and total marks obtained. Develop a program to read data for 5 students and display the overall detail of students rank wise with respect to total marks.

```
#include<stdio.h>
struct std_record
{
    char name[30];
    char faculty[5];
    char email[100];
    int marks;
};
int main()
{
    struct std_record a[5];
    int i=0,high=0;

    for(i=0;i<5;i++)// remember to use fflush(stdin);
    {
        printf("Enter Your Name: ");
        gets(a[i].name);
        fflush(stdin);
        printf("Enter Your Faculty: ");
        scanf("%d",&a[i].faculty);
        fflush(stdin);
        printf("Enter Your Email: ");
        gets(a[i].email);
        fflush(stdin);
        printf("Enter Your Full Marks: ");
        scanf("%d",&a[i].marks);
        fflush(stdin);
    }
    for(i=0;i<5;i++)
    {
        if(high<a[i].marks)
        {
            high=(a[i].marks);
        }
    }
    printf("\nThe Highest Marks is :%d",high);
    for(i=0;i<5;i++)
    {
        if(high==a[i].marks)//
        {
            printf("\nName:");
            puts(a[i].name);
            printf("\nFaculty:");
            puts(a[i].faculty);
            printf("Email:");
            puts(a[i].email);
        }
    }
    return 0;
}
```

2. Write a program, using structure to input records of 5 students. Members include name, roll number, and marks obtained in math, C programming and English. Display the records of students who have passed in C programming.

```
#include<stdio.h>
struct student
{
    char name[40];
    int roll;
    int maths;
    int cprog;
    int eng;
}a[5];
int main()
{
    int i=0;
    for(i=0;i<5;i++)
    {
        printf("Enter a name:");
        fflush(stdin);
        gets(a[i].name);
        printf("Enter roll number:");
        fflush(stdin);
        scanf("%d",&a[i].roll);
        printf("Enter marks of math:");
        fflush(stdin);
        scanf("%d",&a[i].maths);
        printf("Enter marks of c programming:");
        fflush(stdin);
        scanf("%d",&a[i].cprog);
        printf("Enter marks of english:");
        fflush(stdin);
        scanf("%d",&a[i].eng);
        printf("\n");
    }
    printf("The detail of students who passed in c programming:");
    for(i=0;i<5;i++)
    {
        if(a[i].cprog>=32)
        {
            printf("\nName:%s",a[i].name);
            printf("\nRoll number:%d",a[i].roll);
            printf("\nMarks in math:%d",a[i].maths);
            printf("\nMarks in c programming:%d",a[i].cprog);
            printf("\nMarks in english:%d",a[i].eng);
            printf("\n");
        }
    }
    return 0;
}
```

3. Define a structure data type called time-struct containing three members hour, minute and second. Develop a program that would assign values to the individual members and display the time in the form 16:40:51.

```
#include<stdio.h>
struct
{
    int sec;
    int min;
    int hour;
}x;
int main()
{
    int rem;
    printf("Enter second:");
    scanf("%d",&x.sec);
    x.hour=x.sec/3600;
    rem=x.sec%3600;
    x.min=rem/60;
    x.sec=rem%60;
    printf("%d:%d:%d",x.hour,x.min,x.sec);
    return 0;
}
```

4. Write a program to input 5 employee records (Emp_id, Emp_name and Emp_Salary). Display 3 employee information who gets the highest salary.

```
#include <stdio.h>
#include <string.h>
struct emp_records
{
    int emp_id;
    char emp_name[30];
    float emp_salary;
};
int main()
{
    struct emp_records a[5];
    int i=0,j=0;
    for(i=0;i<5;i++)
    {
        printf("Employee number %d\n",i+1);
        printf("Enter Id: ");
        scanf("%d",&a[i].emp_id);
        fflush(stdin);
        printf("Enter Name: ");
        gets(a[i].emp_name);
        fflush(stdin);
        printf("Enter Salary: ");
        scanf("%f",&a[i].emp_salary);
        printf("\n\n");
    }
    struct emp_records highest_salary[3];
    for (i=0;i<3;i++)
    {
        int max=0;
        for (j=1;j<5;j++)
        {
            if (a[j].emp_salary > a[max].emp_salary)
            {
                max=j;
            }
        }
        highest_salary[i] = a[max];
        a[max].emp_salary = -1;
    }
    printf("\nEmployees with the highest salary:\n");
    for (int i = 0; i < 3; i++)
    {
        printf("Employee ID: %d\n", highest_salary[i].emp_id);
        printf("Employee Name: %s\n", highest_salary[i].emp_name);
        printf("Employee Salary: %.2f\n\n", highest_salary[i].emp_salary);
    }
    return 0;
}
```

5. Write a program to input 5 employee records (Emp_id, Emp_name and Emp_Salary). Display employee information whose name starts with 'D'.

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

```
struct emp_record
```

```
{  
    int emp_id;  
    char emp_name[40];  
    int emp_salary;
```

```
};
```

```
int main()
```

```
{
```

```
    struct emp_record record[4];
```

```
    int i=0,h=0;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        printf("\n Enter id:");
```

```
        scanf("%d",&record[i].emp_id);
```

```
        fflush(stdin);
```

```
        printf("\n Enter name:");
```

```
        gets(record[i].emp_name);
```

```
        printf("\n Enter salary:");
```

```
        scanf("%d",&record[i].emp_salary);
```

```
    }
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        if(record[i].emp_name[i]=="d" || "D")
```

```
        {
```

```
            printf("this");
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```


String

1. Write a program to print ASCII values of each and every character of the string given by the user.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[100];
    int i=0;
    printf("Enter the sentence: ");
    gets(a);
    for(i=0;i<strlen(a);i++)
    {
        if(a[i]==32)
        {
            continue;
        }
        printf("The ASCII value of %c is %d",a[i],a[i]);
        printf("\n");
    }
    return 0;
}
```

2. Write a program to reverse a string without using string handling function.

```
#include<stdio.h>
int main()
{
    char word[100],rev[100];
    int i=0,count=0;
    printf("Enter a word: ");
    gets(word);
    for(i=0;word[i]!='\0';i++)
    {
        count++;
    }
    for(i=0;word[i]!='\0';i++)
    {
        rev[i]=word[count-i-1];
    }
    printf("\nReverse of %s word is %s",word,rev);

    return 0;
}
```

3. Write a program to compare whether two strings are identical or not without using string handling function.

```
#include<stdio.h>
int main() {
    char a[10], b[10];
    int i = 0, count1 = 0, count2 = 0, flag = 0;
    kist:
    printf("Enter the first word: ");
    gets(a);
    printf("Enter the second word: ");
    gets(b);

    for (i=0; a[i] != '\0'; i++) //for 1st word
    {
        count1++;
        if (a[i] >= 65 && a[i] <= 90)
        {
            a[i] = a[i] + 32;
        }
    }

    for (i = 0; b[i] != '\0'; i++) //for 2nd word
    {
        count2++;
        if (b[i] >= 65 && b[i] <= 90)
        {
            b[i] = b[i] + 32;
        }
    }
    //this part is activated only when the length of letter of 1st and 2nd word is same

    if (count1 == count2)
    {
        for (i=0; i<count1; i++)
        {
            if (a[i] != b[i])
            {
                flag=2;
                break;
            }
        }
    }
    else //this activates only when the length of letter is different
    {
        printf("Not the same word length\n");
        goto kist; // force jumps to input part
    }

    if (flag == 0)
    {
        printf("\nSame words");
    }
    else
    {
        printf("Not the same words");
    }

    return 0;
}
```

4. Write a program to count no of spaces, vowels in a sentence input by the user.

```
#include<stdio.h>
int main()
{
    char word[100];
    int i=0,vcount=0,space=0;

    printf("Enter a word: ");
    gets(word);
    //for vowel counting
    for(i=0;word[i]!='\0';i++)
    {
        if(word[i]=='a' || word[i]=='A' ||
        word[i]=='e' || word[i]=='E' ||
        word[i]=='i' || word[i]=='I' ||
        word[i]=='o' || word[i]=='O' ||
        word[i]=='u' || word[i]=='U')
        {
            vcount++;
        }
        //for space counting
        if(word[i]==' ') //if(word[i]==32)
        {
            space++;
        }
    }

    printf("\n No of vowels is %d",vcount);
    printf("\n No of spaces is %d",space);
    return 0;
}
```

5. Write a program to convert a lower case string to upper case string without using string handling function.

```
#include<stdio.h>
int main()
{
    char a[100],i=0;
    printf("Enter your word: ");
    gets(a);
    for(i=0;a[i]!='\0';i++)
    {
        if(a[i]>96&&a[i]<123)
        {
            a[i]=a[i]-32 ;
        }
    }
    for(i=0;a[i]!='\0';i++)// or can simply do puts(a)
    {
        printf("%c",a[i]);
    }
    return 0;
}
```

6. Write a program to combine two different words into one without using string handling function (e.g.: word 1: kist, word 2: college, Result: kist college)

```
#include<stdio.h>
int main()
{
    char a[20],b[10];
    int i=0,count=0;
    printf("Enter first word:");
    gets(a);
    printf("Enter second word:");
    gets(b);
    for(i=0;a[i]!='\0';i++)
    {
        count++;
    }
    for(i=0;b[i]!='\0';i++)
    {
        a[count+i]=b[i];
    }
    puts(a);
    return 0;
}
```

Array

1. Write a program to take 10 numbers from user and only display numbers which are prime.

```
#include<stdio.h>
int main()
{
    int i=0,count=0,j=1,a[10];

    for(i=0;i<10;i++)
    {
        printf("Enter number a[%d]:",i);
        scanf("%d",&a[i]);
    }

    for(i=0;i<10;i++)
    {
        count=0;
        for(j=1;j<=a[i];j++)
        {
            if(a[i]%j==0)
            {
                count++;
            }
        }
        if(count==2)
        {
            printf("%d is prime\n",a[i]);
        }
    }
    return 0;
}
```

2. Write a program to take 10 numbers from user and arrange them in reverse order.

```
#include <stdio.h>
int main()
{
    int i=0,temp=0,a[10];
    printf("Enter 10 numbers:\n");
    for (i=0;i<10;i++)
    {
        printf("Enter number %d:",i+1);
        scanf("%d",&a[i]);
    }
    for (i=0;i<5;i++)
    {
        temp=a[i];
        a[i]=a[9-i];
        a[9-i]=temp;
    }
    printf("Reversed:\n");
    for (i=0;i<10;i++)
    {
        printf("%d\n",a[i]);
    }
    printf("\n");
    return 0;
}
```

3. Write a program to take 10 numbers from user and count how many even and odd numbers are present.

```
#include<stdio.h>
int main()
{
    int i=0,count1=0,count2=0,a[10];

    for(i=0;i<10;i++)
    {
        printf("Enter number a[%d]:",i);
        scanf("%d",&a[i]);
    }

    for(i=0;i<10;i++)
    {
        if(a[i]%2==0)
        {
            count1++;
        }
        else
        {
            count2++;
        }
    }
    printf("The number of even are:%d\n and \n\nThe Number of odd are:%d",count1,count2);
    return 0;
}
```

4. Write a program to take 3x3 matrix input from user and calculate sum of diagonal elements.

```
#include<stdio.h>
int main()
{
    int i=0,j=0,sum=0;
    int a[3][3];
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("Enter a[%d][%d] element:",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }
    for (i=0;i<3;i++)
    {
        sum=sum+a[i][i];
    }

    printf("The sum of diagonal elements is: %d\n", sum);
    return 0;
}
```

5. Write a program to take 3x3 matrix input from user and calculate sum of four corners element.

```
#include<stdio.h>
int main()
{
    int a[3][3];
    int sum,i=0,j=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("Enter the element of %d %d:",i,j);
            scanf("%d",&a[i][j]);
            if(i==0&&j==0||i==0&&j==2||i==2&&j==0||i==2&&j==2)
            {
                sum+=a[i][j];
            }
        }
    }
    printf("The sum of 4 corner elements is:%d",sum);
    return 0;
}
```

6. Write a program to take two 3X3 matrix, add them and display transpose of final result.

```
#include <stdio.h>

int main() {
    int matrix1[3][3], matrix2[3][3], sum[3][3], transpose[3][3], i=0, j=0;

    printf("Enter elements of first 3x3 matrix:\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            printf("Enter element [%d][%d]: ", i+1, j+1);
            scanf("%d", &matrix1[i][j]);
        }
    }

    printf("\nEnter elements of second 3x3 matrix:\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            printf("Enter element [%d][%d]:", i+1, j+1);
            scanf("%d", &matrix2[i][j]);
        }
    }

    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            sum[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

    printf("\nSum of the two matrices:\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            printf("%d\t", sum[i][j]);
        }
        printf("\n");
    }

    printf("\nTranspose of the sum matrix:\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            transpose[j][i] = sum[i][j];
        }
    }

    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
        {
            printf("%d\t", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```


7. Write a program to take 10 numbers from user and display the numbers in ascending order, counting total number of digits which are odd.

```
#include<stdio.h>
int main()
{
    int i=0,a[10],temp=0,j=0,oddc=0;
    for(i=0;i<10;i++)
    {
        printf("Enter the Elements a[%d]:",i);
        scanf("%d",&a[i]);
    }
    for(i=0;i<10;i++)
    {
        for(j=0;j<10-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    for(i=0;i<10;i++)
    {
        printf("%d\n",a[i]);
    }
    for(i=0;i<10;i++)
    {
        if((a[i])%2!=0)
        {
            oddc++;
        }
    }
    printf("The Number of Odd Are:%d",oddc);
    return 0;
}
```

Recursion

1. Write a program to find sum of Natural Numbers Using Recursion.

```
#include <stdio.h>
int sum(int);
int main()
{
    int result=0,n=0;
    printf("Enter value of n:");
    scanf("%d",&n);
    result=sum(n);
    printf("sum = %d",result);
    return 0;
}
int sum(int n)
{
    int b;

    while(m!=0)
    {
        b=m+sum(m-1);
        return b;
    }
    while(m==0)
    {
        return 0;
    }

}
```

2. Write a program to generate Fibonacci Series using recursion.

```
#include<stdio.h>
int fibonacci(int);
int main()
{
    int i=0,n=0;
    printf("Enter n:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("%d\t",fibonacci(i));
    }
    return 0;
}
int fibonacci(int n)
{
    int b;
    if(n==0)
        return 0;

    else if(n==1)
        return 1;

    else
        b=(fibonacci(n-1)+(fibonacci(n-2)));
    return b;
}
```

3. Write a program to find the power of a given number using recursion.

```
#include<stdio.h>
int power(int,int);
int main()
{
    int base,p,result;
    printf("Enter Base: ");
    scanf("%d",&base);
    printf("Enter Power: ");
    scanf("%d",&p);
    printf("%d^%d=%d",base,p,power(base,p));
    return 0;
}
int power(int base,int p)
{
    if(p!=0)
        return(base*power(base,p-1));
    else
        return 1;
}
```

4. Write a program to find factorial of a given number using recursion.

```
#include <stdio.h>
int factorial(int);
int main()
{
    int result=0,n=0;
    printf("Enter value of n:");
    scanf("%d",&n);
    result=factorial(n);
    printf("factorial = %d",result);
    return 0;
}
int factorial(int n)
{
    if (n==0)
        return 1;
    else
        return n*factorial(n-1);
}
```

DMA

1. Write a program using DMA to take 10 numbers from user and only display numbers which aren't prime.

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

int main()
{
    int n=0,i=0,count=0;
    int *p;

    printf("Enter how many numbers you want: ");
    scanf("%d", &n);

    p=(int*)calloc(n, sizeof(int));
    if(p==NULL)
    {
        printf("Memory not allocated.");
        exit(0);
    }

    for (i=0;i<n;i++)
    {
        printf("Enter number: ");
        scanf("%d", p + i);
    }

    printf("Numbers with even indices and value of each number: ");
    for (i = 0; i < n; i++)
    {
        if (*(p + i) % 2 == 0)
        {
            count++;
            if (count >= 2)
            {
                printf("%d\t", *(p + i));
            }
        }
    }

    free(p);
    return 0;
}
```

2. Write a program using DMA to take 10 numbers from user and arrange them in reverse order.

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

```
int main()
{
    int *p;
    int i=0;
    p=(int*)calloc(10,sizeof(int));
    if (p==NULL)
    {
        printf("Memory allocation failed.\n");
        exit(0);
    }
    printf("Enter 10 numbers:\n");
    for (i=0;i<10;i++)
    {
        scanf("%d",&p[i]);
    }
    printf("\n In reverse order:\n");
    for (i=9;i>=0;i--)
    {
        printf("%d\n",p[i]);
    }
    printf("\n");
    free(p);

    return 0;
}
```

3. Write a program using DMA to take 10 numbers from user and find sum of all 10 digits, and check whether sum is palindrome or not.

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

int main()
{
    int *n;
    int sum=0, rev=0, temp=0;
    int i;
    n=(int*)calloc(10, sizeof(int));
    if(n==NULL)
    {
        printf("Memory allocation failed.\n");
        exit(0);
    }
    printf("Enter 10 numbers:\n");
    for (i=0; i<10; i++)
    {
        scanf("%d", &n[i]);
        {
            sum=sum+n[i];
        }
    }
    temp=sum;
    while(temp!=0)
    {
        rev=rev*10+temp%10;
        temp=temp/10;
    }
    if(sum==rev)
    {
        printf("The sum %d is a palindrome\n", sum);
    }
    else
    {
        printf("The sum %d is not a palindrome\n", sum);
    }
    free(n);

    return 0;
}
```

4. Write a program using DMA to take 10 numbers from user and find sum of all 10 digits, and check whether sum is Armstrong or not.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int *n;
    int i=0, sum=0, temp=0, temp2=0, r=0, d=0;
    n=(int*)calloc(10, sizeof(int));
    if(n==NULL)
    {
        printf("No memory allocation");
        exit(0);
    }
    printf("Enter 10 numbers:\n");
    for (i=0; i<10; i++)
    {
        scanf("%d", &n[i]);
        sum=sum+n[i];
    }
    temp=sum;
    temp2=sum;
    while(temp!=0)
    {
        d=temp%10;
        r=pow(d, 3)+r;
        temp=temp/10;
    }
    if(sum==temp2)
    {
        printf("The sum %d is an Armstrong number\n", sum);
    }
    else
    {
        printf("The sum %d is not an Armstrong number\n", sum);
    }
    free(n);
    return 0;
}
```

Pointers

1. Write a program to add all the even numbers in an array and display result using pointers only.

```
#include <stdio.h>
int main()
{
    int a[100], i=0, sum=0;
    int *p, *q;
    p=&sum;
    q=a;
    printf("Enter 10 number:\n ");
    for(i=0; i<10; i++)
    {
        printf("Enter a[%d]:", i);
        scanf("%d", &*(q+i));
    }
    for(i=0; i<10; i++)
    {
        if(*(q+i)%2==0) // a[i] is same as *(q+i) || *(a+i) +i means to go to the next
        block of array
        {
            *p=*p+a[i];
        }
    }
    printf("The Sum of All Even Number is :%d", *p);
    return 0;
}
```

2. Write a program to find highest value amongst 10 numbers given by user using pointers.

```
#include<stdio.h>
int main()
{
    int a[10], i=0, gt=0;
    int *p, *q;
    p=a;
    q=&gt;
    printf("Enter 10 Numbers:\n");
    for(i=0; i<10; i++)
    {
        printf("Enter a[%d]:", i);
        scanf("%d", &*(p+i));
    }
    for(i=0; i<10; i++)
    {
        if(*q<*(p+i))
        {
            *q=*(p+i);
        }
    }
    printf("The Greatest Number is: %d", *q);
    return 0;
}
```


3. Write a program to enter 10 numbers and calculate even and odd numbers count using pointers only.

```
#include<stdio.h>
int main()
{
    int a[10],i=0,even_cnt=0,odd_cnt=0;
    int *p,*q,*r;
    p=a;
    q=&even_cnt;
    r=&odd_cnt;
    printf("Enter 10 Numbers:\n");
    for(i=0;i<10;i++)
    {
        printf("Enter a[%d]:",i);
        scanf("%d",&(p+i));
    }
    for(i=0;i<10;i++)
    {
        if(*(p+i)%2==0)
        {
            (*q)++;
        }
        else
        {
            (*r)++;
        }
    }
    printf("The Number of Even is:%d\n",*q);
    printf("The Number of Odd is:%d",*r);

    return 0;
}
```

4. Write a program to add two arrays and count number of odd numbers in the final result using pointers only.

```
#include<stdio.h>
int main()
{
    int a[5],b[5],c[5],i=0,odd_cnt=0;
    int *p,*q,*r,*s;
    p=a;
    q=b;
    r=c;
    s=&odd_cnt;
    printf("Enter 5 numbers for first array\n");
    for(i=0;i<5;i++)
    {
        printf("Enter a[%d]:",i);
        scanf("%d",&*(p+i));
    }
    printf("Enter 5 numbers for second array\n");
    for(i=0;i<5;i++)
    {
        printf("Enter b[%d]:",i);
        scanf("%d",&*(q+i));
    }
    for(i=0;i<5;i++)
    {
        *(r+i)=*(p+i)+*(q+i);
    }
    for(i=0;i<5;i++)
    {
        printf("\n%d+%d=%d",*(p+i),*(q+i),*(r+i));
    }
    for(i=0;i<5;i++)
    {
        if(c[i]%2!=0)
        {
            (*s)++;
        }
    }
    printf("\nThe Number of odd are:%d",*s);
    return 0;
}
```

Function

1. Write a program using function to check whether number is prime or not. (WP,WR).

```
#include <stdio.h>
int prime(int);

int main()
{
    int result=0,n=0;
    printf("\nEnter your number: ");
    scanf("%d",&n);

    result=prime(n);
    if (result == 1)
    {
        printf("\nThe number is prime.\n");
    }
    else
    {
        printf("The number is not prime.\n");
    }
    return 0;
}

int prime(int x)
{
    int i=0;
    if (x<=1)
    {
        return 0;
    }
    for (i=2;i*i<=x;i++)
    {
        if (x%i== 0)
        {
            return 0;
        }
    }
    return 1;
}
```

2. Write a program using function to swap two numbers by passing pointers(WP,WR).

```
#include<stdio.h>
int swap(int*,int*);
int main()
{
    int a=0,b=0,result=0;
    printf("Enter Number a:");
    scanf("%d",&a);
    printf("Enter Number b:");
    scanf("%d",&b);
    result=swap(&a,&b);
    {
        printf("Swapping.....\n");
        printf("a=%d\nb=%d",a,b);
    }
    return 0;
}
int swap(int *p,int *q)
{
    int temp=0;
    if(*p!=*q)
    {
        temp=*p;
        *p=*q;
        *q=temp;
    }
    return *p,*q;
}
```

3. Write a program using function to check whether number is Armstrong or not by passing pointers. (WP,WR)

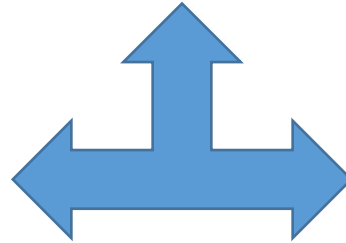
```
#include <stdio.h>
#include <math.h>

int ArmstrongHora(int *num);
int main()
{
    int num;
    printf("Enter a number:");
    scanf("%d",&num);
    if (ArmstrongHora(&num))
    {
        printf("%d is an Armstrong number",num);
    }
    else
    {
        printf("%d is not an Armstrong number",num);
    }
    return 0;
}
int ArmstrongHora(int *num)
{
    int ogNum=0,remainder=0,result=0,n=0;
    ogNum=*num;
    while(ogNum!=0)
    {
        ogNum=ogNum/10;
        ++n;
    }
    ogNum=*num;
    while (ogNum!=0)
    {
        remainder=ogNum%10;
        result=pow(remainder,n)+result;
        ogNum=ogNum/10;
    }
    if(result==*num)
        return 1;
    else
        return 0;
}
```

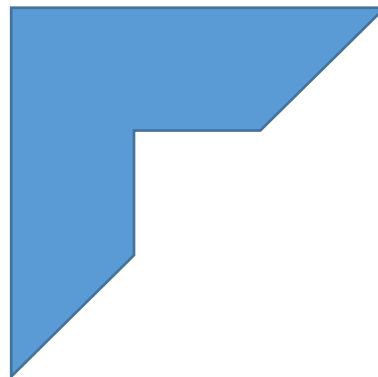
Graphics

```
1.
#include<stdio.h>
#include<graphics.h>
#include<dos.h>

int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    line(150,250,200,200);
    line(200,200,200,225);
    line(150,250,200,300);
    line(200,300,200,275);
    line(200,275,400,275);
    line(400,275,400,300);
    line(400,300,450,250);
    line(450,250,400,200);
    line(400,200,400,225);
    line(400,225,325,225);
    line(325,225,325,150);
    line(325,150,350,150);
    line(350,150,300,100);
    line(300,100,250,150);
    line(250,150,275,150);
    line(275,150,275,225);
    line(275,225,200,225);
    delay(5000);
    closegraph();
    return 0;
}
```

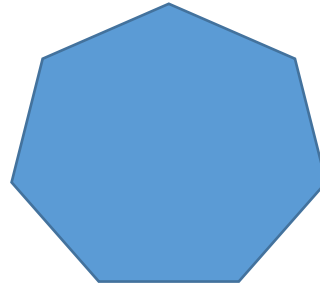


```
2.
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    line(200,100,200,400);
    line(200,400,300,300);
    line(300,300,300,200);
    line(300,200,400,200);
    line(400,200,500,100);
    line(500,100,200,100);
    delay(5000);
    closegraph();
    return 0;
}
```



3.

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    line(300,100,400,150);
    line(400,150,425,250);
    line(425,250,375,350);
    line(375,350,225,350);
    line(225,350,175,250);
    line(175,250,200,150);
    line(200,150,300,100);
    delay(5000);
    closegraph();
    return 0;
}
```



4.

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    rectangle(200,200,500,400);
    line(200,200,300,100);
    line(300,100,600,100);
    line(600,100,500,200);
    line(500,200,500,400);
    line(500,400,600,300);
    line(600,300,600,100);
    delay(5000);
    closegraph();
    return 0;
}
```



5.

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    line(200,100,450,100);
    line(450,100,450,150);
    line(450,150,500,150);
    line(500,150,500,250);
    line(500,250,450,250);
    line(450,250,450,300);
    line(450,300,200,300);
    line(200,300,200,250);
    line(200,250,150,250);
    line(150,250,150,150);
    line(150,150,200,150);
    line(200,150,200,100);
    delay(5000);
    closegraph();
    return 0;
}
```



6.

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"");
    line(200,100,300,100);
    line(300,100,300,350);
    line(300,350,200,350);
    line(200,350,200,250);
    line(200,250,175,250);
    line(175,250,175,275);
    line(175,275,125,275);
    line(125,275,125,175);
    line(175,175,175,200);
    line(175,200,200,200);
    line(200,200,200,100);
    delay(5000);
    closegraph();
    return 0;
}
```

