**Experiment no – 01(**a**)**

**Aim: Write an algorithm and draw flowchart for Area of circle.**

**Algorithm:**

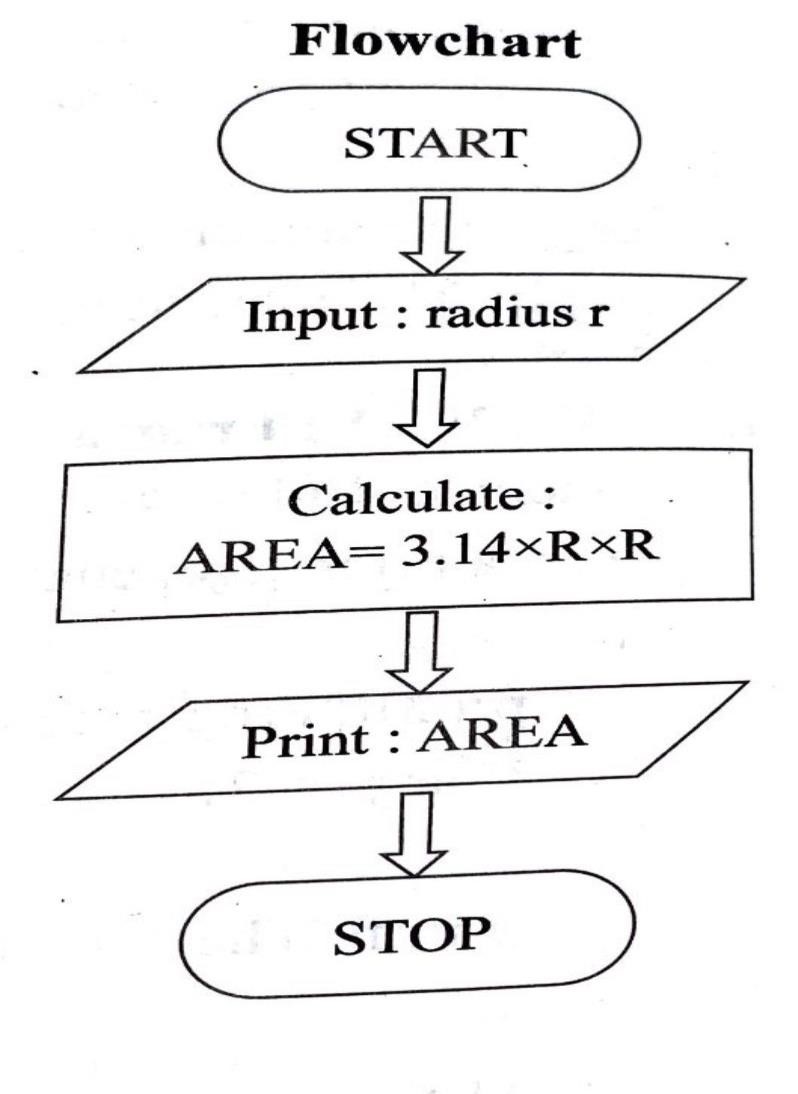
Step 1:Start

Step 2: Read the circle’s radius R value.

Step 3: Calculate area oof circle i.e. AREA= 3.14 x R x R

Step 4: Print AREA

Step 5: Stop **Flowchart:**



**Figure-1**

**Conclusion :** Successfully Drawn flowchart and wrote an algorithm

**Experiment no – 01(b)**

**Aim: Write an algorithm and draw flowchart to print the given no. is even or odd.**

**Algorithm:**

Step 1: Start

Step 2: Read the number value NUM.

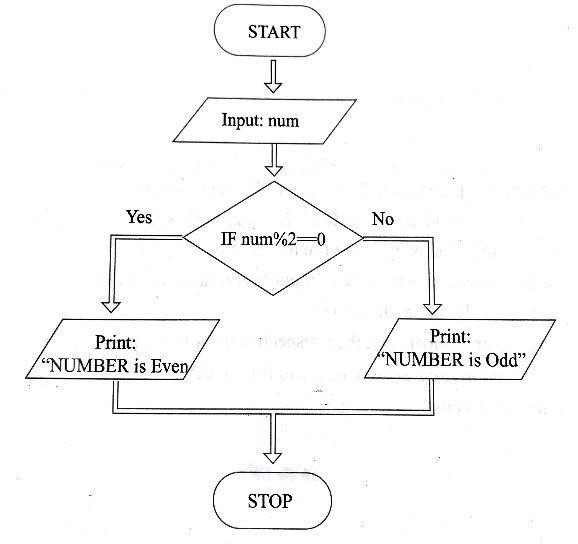
Step 3: Divide the NUM by 2 and store the remainder in REM

Step 4: If REM = 0 Then go to Step 6

Step 5: Print “NUMBER is Odd” go to step 7

Step 6: Print “NUMBER is Even”

Step 7: Stop **Flowchart:**



**Figure-2**

**Conclusion :** Successfully Drawn flowchart and wrote an algorithm.

**Experiment no – 01(**c**)**

**Aim: Write an algorithm and draw flowchart to print 1 to 10 numbers.**

**Algorithm:**

Step 1: Start

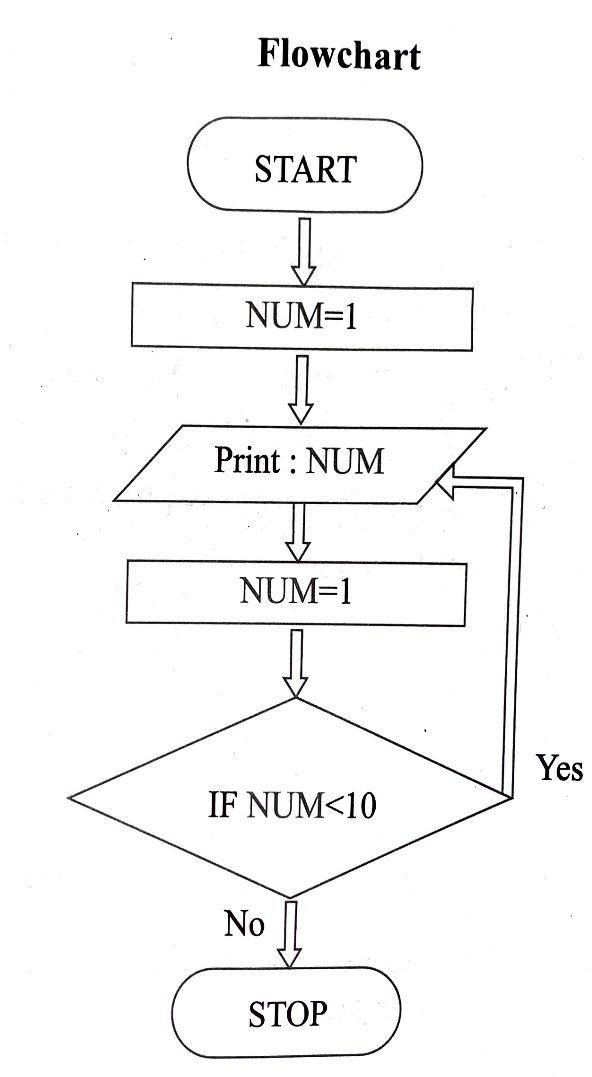
Step 2: Initialize the variable NUM = 1

Step 3: Print NUM

Step 4: Increment NUM by 1 NUM=NUM+1

Step 5: If NUM<= 10 go to Step 3

Step 6: Stop **Flowchart:**



**Figure-3**

**Conclusion :** Successfully Drawn flowchart and wrote an algorithm.

**Experiment no – 01(**d**)**

**Aim: Write an algorithm and draw flowchart for sum of 1 to 5 numbers.**

**Algorithm:**

Step 1: Start

Step 2: Initialize the variable NUM = 1 and SUM=0

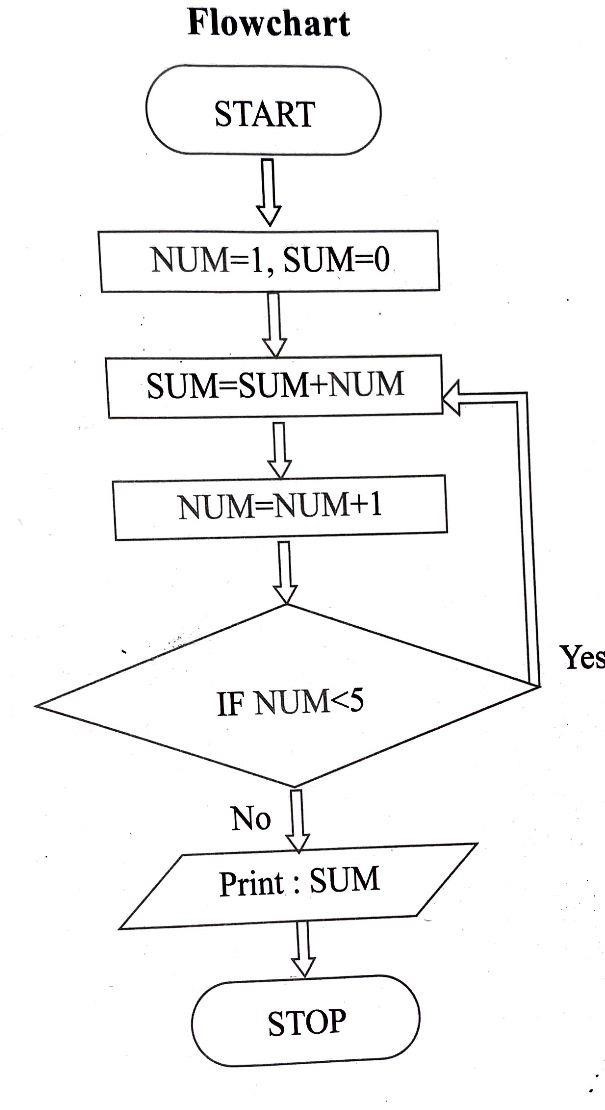
Step 3: SUM=SUM+NUM

Step 4: Increment NUM by 1 NUM=NUM+1

Step 5: If NUM<=5 go to Step 3

Step 6: Print SUM

Step 7: Stop **Flowchart:**



**Figure-4**

**Conclusion :** Successfully Drawn flowchart and wrote an algorithm.

**Experiment no – 01(**e**)**

**Aim: Write an algorithm and draw flow chart to compute the addition of digits of a given number.**

**Algorithm:**

Step 1: Start

Step 2: Read the number value NUM

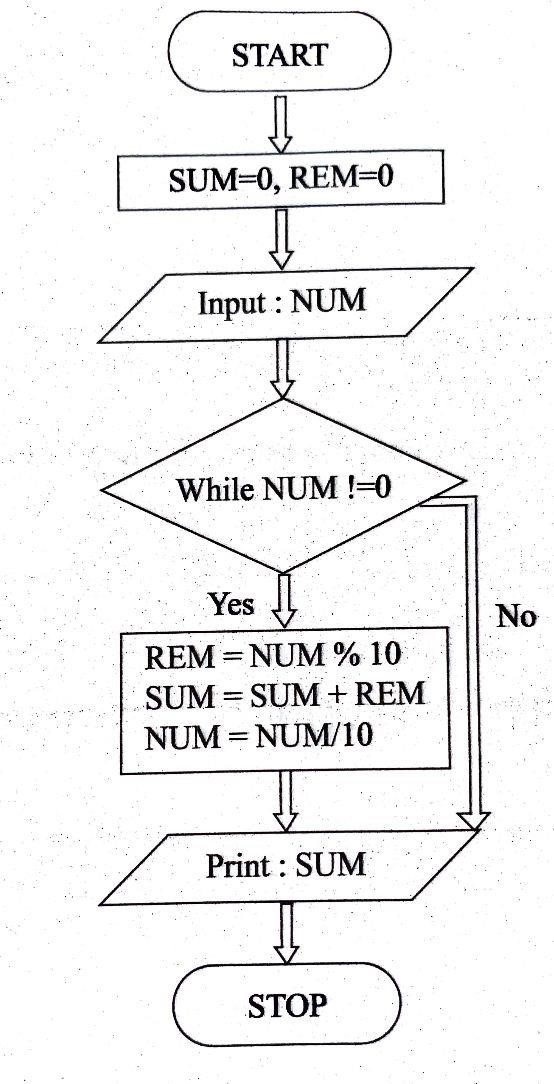
Step 3: Initialize SUM = 0

Step 4: Perform REM = NUM % 10 and add REM to SUM i.e. SUM = SUM + REM

Step 5: Perform NUM = NUM/10

Step 6: IF NUM = 0 stop the process and Print SUM else go to Step 3

Step 7: Stop **Flowchart:**



**Figure-5**

**Conclusion :** Successfully Drawn flowchart and wrote an algorithm.

**Experiment no – 02(a)**

**Aim: Write a program using while loop to reverse the digits of a number.**

**Algorithm:**

1. Ask the user to enter any number.
2. Declare and initialize another variable reversed with 0, where reversed an integer variable.
3. Get the last digit of the given number by performing the modulo division (%) and store the value in last\_digit variable, likey last\_digit= number % 10.
4. Multiply reversed by 10 and add last\_digit, like reversed = reversed\*10 + last\_digit.
5. Divide numbered by 10, like numbered/10.
6. Repeat the steps 3 to 5 till numbered is not equal to (or greater than) zero.

**Code**:

*#include <stdio.h>*

*int main() {*

*printf("01-AlstonAlvares."); int num, rnum = 0, rem;*

*printf("Enter any number: ");*

*scanf("%d", &num);*

*while (num != 0) {*

*rem = num % 10;*

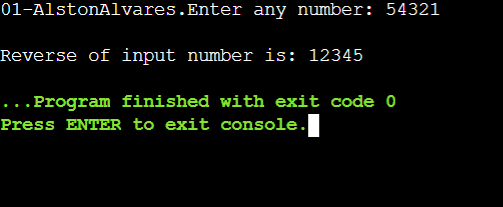
*rnum = rnum \* 10 + rem;*

*num = num / 10; }*

*printf("\nReverse of input number is: %d", rnum);*

*return 0;}*

**Output:**

****

**Figure-6**

**Conclusion :** Successfully performed program using while loop to reverse the digits of a number.

**Experiment no – 02(b)**

**Aim: Write a program to calculate the factorial of a given number.**

**Algorithm:**

1. Start program
2. Ask the user to enter an integer to find the factorial
3. Read the integer and assign it to a variable
4. From the value of the integer up to 1, multiply each digit and update the final value
5. The final value at the end of all the multiplication till 1 is the factorial
6. End program

**Code:**

*#include <stdio.h>*

*int main() {*

*{*

*printf("01-AlstonAlvares.");}*

*int n, i;*

*unsigned long long fact = 1;*

*printf("Enter an integer: ");*

*scanf("%d", &n);*

*// shows error if the user enters a negative integer*

*if (n < 0)*

*printf("Error! Factorial of a negative number doesn't exist.");*

*else {*

*for (i = 1; i <= n; ++i) {*

*fact \*= i;*

*}*

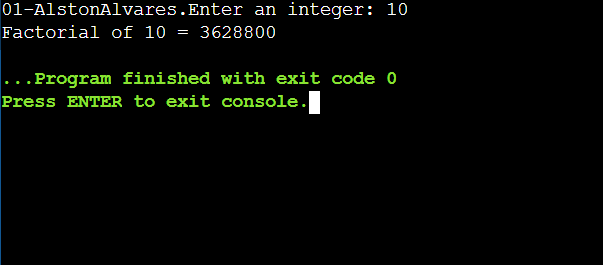
*printf("Factorial of %d = %llu", n, fact);*

*}*

*return 0;*

*}*

**Output:**



**Figure-7**

**Conclusion :** Successfully performed a program to calculate the factorial of a given number.

**Experiment no – 02(c)**

**Aim: Write a program to find the roots of quadratic equation.**

**Algorithm:**

* 1. Start
  2. Read a, b, c values
  3. Compute d = b2 4ac
  4. if d > 0 then
     + 1. r1 = b+ sqrt (d)/(2\*a)
       2. r2 = b sqrt(d)/(2\*a)
  5. Otherwise if d = 0 then
     + 1. compute r1 = -b/2a, r2=-b/2a
       2. print r1,r2 values
  6. Otherwise if d < 0 then print roots are imaginary
  7. Stop

**Code:**

*#include<stdio.h>*

*#include<math.h>*

*int main()*

*{*

*printf("01-AlstonAlvares.");*

*float a,b,c,x1,x2,determinant,realpart,imaginaryPart;*

*printf("Enter coefficients a,b and c:");*

*scanf("%f%f%f",&a,&b,&c);*

*determinant=b\*b - 4\*a\*c;*

*if (determinant>0)*

*{*

*x1=(-b + sqrt(determinant))/(2\*a);*

*x2=(-b - sqrt(determinant))/(2\*a);*

*printf("Roots are real and different.");*

*printf("\n x1=%.3f",x1);*

*printf("\n x2=%.3f",x2);*

*}*

*else if (determinant==0)*

*{*

*printf("Roots are real and same.");*

*x1=(-b+sqrt(determinant))/(2\*a);*

*printf("\n x1=%.ef",x1);*

*printf("\nx2=%.3f",x2);*

*}*

*Else*

*{*

*realpart=-b/(2\*a);*

*imaginaryPart=sqrt(determinant)/(2\*a);*

*printf("\n Roots are complex and differtent.");*

*printf("\n x1=%.3f+%.fi",realpart,imaginaryPart);*

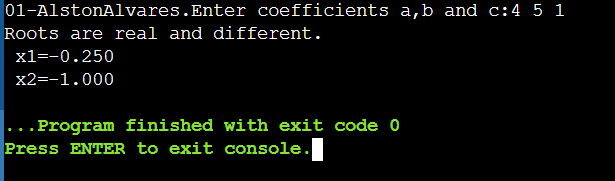
*printf("\nx2 = %.3f-%3fi",realpart,imaginaryPart);*

*}*

*return 0;*

*}*

**Output:**



**Figure-8**

**Conclusion :** Successfully performed a program to find the roots of quadratic equation**.**

**Experiment no – 02(d)**

**Aim: Write a program to print the Fibonacci series.**

**Algorithm:**

1. START
2. Take integer variable A, B, C
3. Set A = 0, B = 0
4. DISPLAY A, B
5. C = A + B
6. DISPLAY C
7. Set A = B, B = C
8. REPEAT from 4 - 6, for n times
9. STOP

**Code:**

*#include <stdio.h>*

*int main() {*

*printf("01-AlstonAlvares.");*

*int i, n;*

*// initialize first and second terms*

*int t1 = 0, t2 = 1;*

*// initialize the next term (3rd term)*

*int nextTerm = t1 + t2;*

*// get no. of terms from user*

*printf("Enter the number of terms: ");*

*scanf("%d", &n);*

*// print the first two terms t1 and t2*

*printf("Fibonacci Series: %d, %d, ", t1, t2);*

*// print 3rd to nth terms*

*for (i = 3; i <= n; ++i) {*

*printf("%d, ", nextTerm);*

*t1 = t2;*

*t2 = nextTerm;*

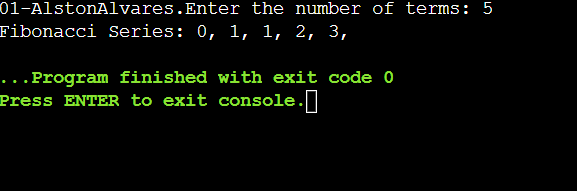
*nextTerm = t1 + t2;*

*}*

*return 0;*

*}*

**Output:**

****

**Figure-9**

**Conclusion :** Successfully performed a program to print the Fibonacci series

**Experiment no – 03(a)**

**Aim: Write a program in C to check entered character vowel or consonant.**

**Algorithm:**

1. Start
2. Declare character type variable ch
3. Read ch from User
4. // Checking both lower and upper case vowels.
5. IF (ch == 'a' || ch == 'A' ||
   * 1. ch == 'e' || ch == 'E' ||
     2. ch == 'i' || ch == 'I' ||
     3. ch == 'o' || ch == 'O' ||
     4. ch == 'u' || ch == 'U' )
6. Print "Vowel"
7. ELSE
8. Print "Consonant"
9. Stop

**Code:**

*#include <stdio.h>*

*int main() {*

*char c;*

*printf("01-AlstonAlvares.");*

*int lowercase\_vowel, uppercase\_vowel;*

*printf("Enter an alphabet: ");*

*scanf("%c", &c);*

*// evaluates to 1 if variable c is a lowercase vowel*

*lowercase\_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');*

*// evaluates to 1 if variable c is a uppercase vowel*

*uppercase\_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');*

*// evaluates to 1 (true) if c is a vowel*

*if (lowercase\_vowel || uppercase\_vowel)*

*printf("%c is a vowel.", c);*

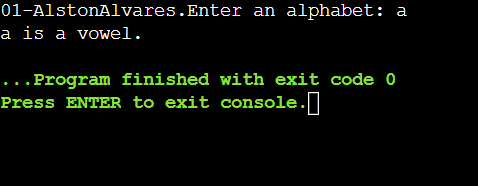
*else*

*printf("%c is a consonant.", c);*

*return 0;*

*}*

**Output:**



**Figure-10**

**Conclusion :** Successfully performed **a** program in C to check entered character vowel or consonant.

**Experiment no – 03(b)**

**Aim:** **Write a program to C program to print day name of week using switch-case.**

**Algorithm:**

1. Input day number from user. Store it in some variable say *no*.
2. Switch the value of *week* i.e. use switch(no) and match with cases.
3. There can be 7 possible values(choices) of *week* i.e. 1 to 7. Therefore write 7 case inside switch. In addition, add default case as an else block.
4. For case 1: print “MONDAY”, for case 2: print “TUESDAY” and so on. Print “SUNDAY” for case 7:.
5. If any case does not matches then, for default: case print “Invalid week number”.

**Code:**

*#include <stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.");*

*int week;*

*/\* Input week number from user \*/*

*printf("Enter week number(1-7): ");*

*scanf("%d", &week);*

*switch(week)*

*{*

*case 1:*

*printf("Monday");*

*break;*

*case 2:*

*printf("Tuesday");*

*break;*

*case 3:*

*printf("Wednesday");*

*break;*

*case 4:*

*printf("Thursday");*

*break;*

*case 5:*

*printf("Friday");*

*break;*

*case 6:*

*printf("Saturday");*

*break;*

*case 7:*

*printf("Sunday");*

*break;*

*default:*

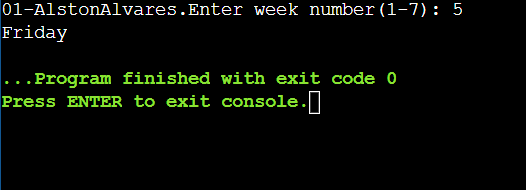
*printf("Invalid input! Please enter week number between 1-7.");*

*}*

*return 0;*

*}*

**Output:**



**Figure-11**

**Conclusion :** Successfully performed a program to C program to print day name of week using switch-case

**Experiment no – 03(c)**

**Aim:** **Write a program to read three values from keyboard and print out the largest of them without using if statement.**

**Algorithm:**

1. Ask the user to enter three integer values.
2. Read the three integer values in num1, num2, and num3 (integer variables).
3. Check if num1 is greater than num2.
4. If true, then check if num1 is greater than num3.
5. If false, then check if num2 is greater than num3.

**Code:**

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.");*

*int N1, N2, N3, Irg; printf("Enter three numbers:");*

*scanf("%d %d %d", &N1, &N2, &N3);*

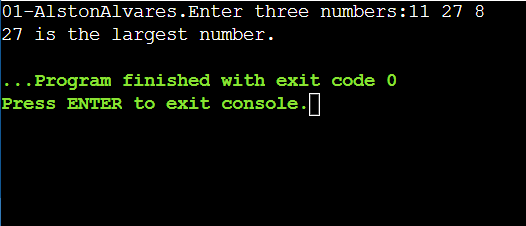
*Irg = N1 > N2 ? (N1 > N3 ? N1 : N3) : (N2 > N3 ? N2 : N3);*

*printf("%d is the largest number.",Irg);*

*return 0;*

*}*

**Output:**



**Figure-12**

**Conclusion :** Successfully performed a program to read three values from keyboard and print out the largest of them without using if statement.

**Experiment no – 04(a)**

**Aim**: **a. Write a program to print the pattern of asterisks as shown below :**

\*

\* \*

\* \* \*

\* \* \* \*

**Algorithm:**

Display \* and go to new line

Step 2: Display \* \* and go to new line.

Step 3: Display \* \* \* and go to new line.

Step 4: Display \* \* \* \*

Display \* and go to new line

Step 2: Display \* \* and go to new line.

Step 3: Display \* \* \* and go to new line.

Step 4: Display \* \* \* \*

Display \* and go to new line

Step 2: Display \* \* and go to new line.

Step 3: Display \* \* \* and go to new line.

Step 4: Display \* \* \* \*

1. Display \* and go to new line
2. Display \* \* and go to new line.
3. Display \* \* \* and go to new line.
4. Display \* \* \* \*

**Code:**

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.\n");*

*int i, j, n;*

*/\* for used as row wise \*/*

*for(i=1; j<=4; ++i)*

*{ /\* for used as column wise \*/*

*for(j=1; j<=i; ++j)*

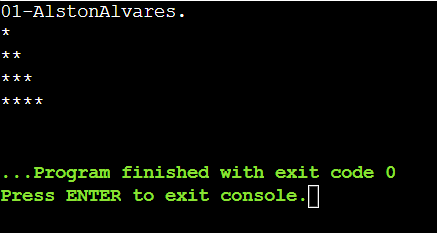
*{*

*printf("\*"); } printf("\n");*

*}*

*return 0;}*

**Output:**



**Figure-13**

**Conclusion :** Successfully performed a program to print the pattern of asterisks.

**Experiment no – 04(b)**

**Aim: Write a program to print the pattern of asterisks as shown below :**

**\* \* \* \* \***

**\* \* \* \***

**\* \* \***

**\* \***

**\***

**Algorithm:**

1. Display \*\*\*\*\* and go to new line
2. Display \* \* \*\*and go to new line.
3. Display \* \* \* and go to new line.
4. Display \* \* and go to new line.
5. Display \*

**Code:**

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.\n");*

*int i, j; /\* for used as row wise \*/*

*for(i=5; i>=1; i--)*

*{* */\* for used as column wise \*/*

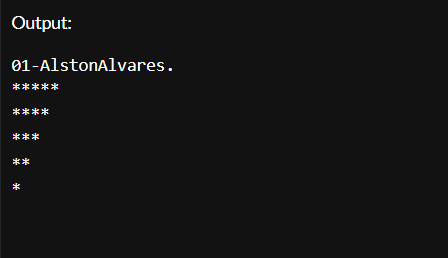
*for(j=1; j<=i; j++) {*

*printf("\*"); }*

*printf("\n"); }*

*return 0; }*

**Output:**



**Figure-14**

**Conclusion :** Successfully performed a program to print the pattern of asterisks.

**Experiment no – 04(c)**

**Aim: Write a program to print Floyd’s Triangle.**

**Algorithm:**

1. Create variables that hold rows and column values as i and j. Take a number to display the rows as num and set the variable k to 1as its initial value.
2. Use nested for loops:
   1. Outer for loop starts its iteration i = 1 up to n rows.
   2. Inner for loop starts its iteration from j = 1 up to (j <=i).
3. Print the values of k.
4. Increment k by 1 or k = k + 1.
5. Jump to newline after each iteration of the inner for loop.
6. Stop

**Code:**

*#include <stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int n, i, c, a = 1;*

*printf("Enter the number of rows of Floyd's triangle to print\n");*

*scanf("%d", &n);*

*for (i = 1; i <= n; i++)*

*{*

*for (c = 1; c <= i; c++)*

*{*

*printf("%d ", a); // Please note space after %d*

*a++;*

*}*

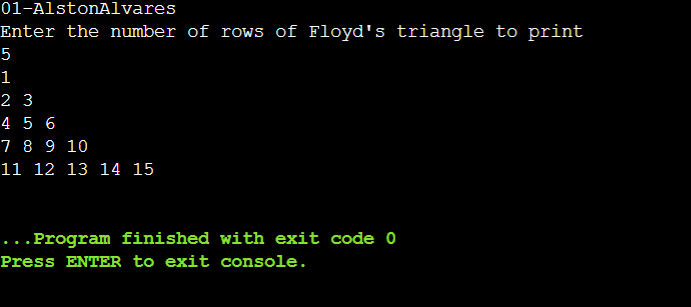
*printf("\n");*

*}*

*return 0;*

*}*

**Output:**



**Figure-15**

**Conclusion :** Successfully performed a program to print Floyd’s Triangle.

**Experiment no – 05(a)**

**Aim: Write a program to print area of square using function.**

**Algorithm:**

1. Start.
2. Declare at s as integer.
3. Initialize value of s.
4. Calculate at s×s.
5. print area of triangle .
6. End.

**Code:**

*#include <stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

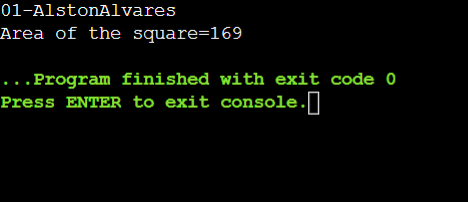
*int s=13;*

*int area\_square=s\*s;*

*printf("Area of the square=%d",area\_square);*

*}*

**Output:**



**Figure-16**

**Conclusion :** Successfully performed a program to print area of square using function.

**Experiment no – 05(b)**

**Aim: Write a program using recursive function.**

**Algorithm:**

1. Start.
2. Read the Input.
3. Perform recursion.
4. Print result.
5. Stop.

**Code:**

*#include <stdio.h>*

*int sum(int n);*

*int main() { printf("01-AlstonAlvares.\n");*

*int number, result;*

*printf("Enter a positive integer: ");*

*scanf("%d", &number);*

*result = sum(number);*

*printf("sum = %d", result);*

*return 0;*

*}*

*int sum(int n) {*

*if (n != 0)*

*// sum() function calls itself*

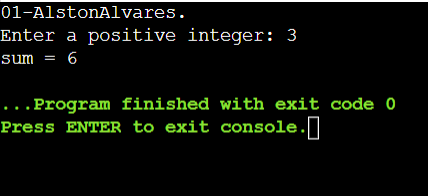
*return n + sum(n-1);*

*else*

*return n;*

*}*

**Output:**



**Figure-17**

**Conclusion :** Successfully performed a program using recursive function.

**Experiment no – 05(c)**

**Aim: Write a program to square root, abs() value using function.**

**Algorithm:**

1. Start
2. Read the input
3. Calculate absolute value
4. Calculate square root value
5. Print results
6. Stop

**Code:**

*#include<stdio.h>*

*#include<math.h>*

*int main()*

*{ printf("01-AlstonAlvares.\n");*

*int num, a;*

*printf("Please enter a number :\n");*

*scanf("%d",&num);*

*a = abs(num);*

*printf("Calculated absolute value is : %d\n", a);*

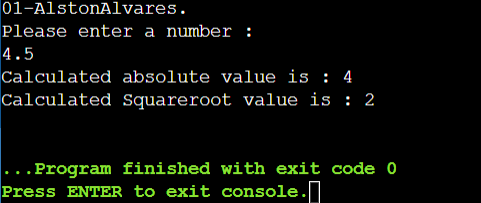
*a = sqrt(num);*

*printf("Calculated Squareroot value is : %d\n",a);*

*return 0;*

*}*

**Output:**



**Figure-18**

**Conclusion :** Successfully performed a program to square root, abs() value using function.

**Experiment no – 05(d)**

**Aim: Write a program using go to statement.**

**Algorithm:**

1. Start
2. Read the Input
3. Check if the input is inside loop or outside loop
4. Print result
5. Stop

**Code:**

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.\n");*

*int n;*

*for(;;) /\*ifinite loop\*/*

*{*

*printf("enter any number :");*

*scanf("%d",&n);*

*if(n == 5)*

*goto ap; /\* use of goto statement\*/*

*if (n% 2 == 0)*

*continue; /\*use of continue statement\*/*

*if (n% 3 == 0)*

*break; /\*use of break state\*/*

*printf("Inside loop");*

*}*

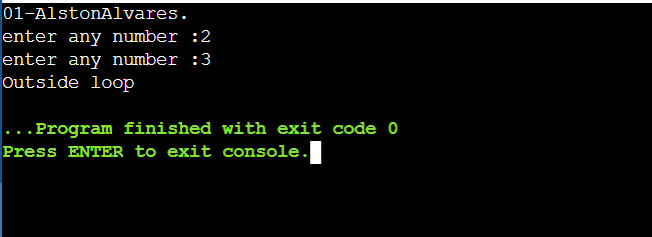
*ap:*

*printf("Outside loop");*

*return 0;*

*}*

**Output:**



**Figure-19**

**Conclusion :** Successfully performed a program using go to statement.

**Experiment no – 06(a)**

**Aim:** **a. Write a program to print rollno and names of 10 students using array.**

**Algorithm:**

1. Start
2. Store Student Information
3. Create the student's structure variable
4. Display information
5. Stop

**Code:**

*// C Program to Store Information*

*// of Students Using Structure*

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <string.h>*

*// Create the student structure*

*struct Student {*

*char\* name;*

*int roll\_number;*

*};*

*// Driver code*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int i = 0, n = 10;*

*// Create the student's structure variable*

*// with n Student's records*

*struct Student student[n];*

*// Get the students data*

*student[0].roll\_number = 1;*

*student[0].name = "Geeks16";*

*student[1].roll\_number = 2;*

*student[1].name = "Geeks54";*

*student[2].roll\_number = 3;*

*student[2].name = "Geeks22";*

*student[3].roll\_number = 4;*

*student[3].name = "Geeks41";*

*student[4].roll\_number = 5;*

*student[4].name = "Geeks39";*

*student[5].roll\_number = 6;*

*student[5].name = "Geeks3”;*

*student[6].roll\_number = 7;*

*student[6].name = "Geeks32";*

*student[7].roll\_number = 8;*

*student[7].name = "Geeks36";*

*student[8].roll\_number = 9;*

*student[8].name = "Geeks35";*

*student[9].roll\_number = 10;*

*student[9].name = "Geeks34";*

*// Print the Students information*

*printf("Student Records:\n\n");*

*for (i = 0; i < n; i++) {*

*printf("\tName = %s\n", student[i].name);*

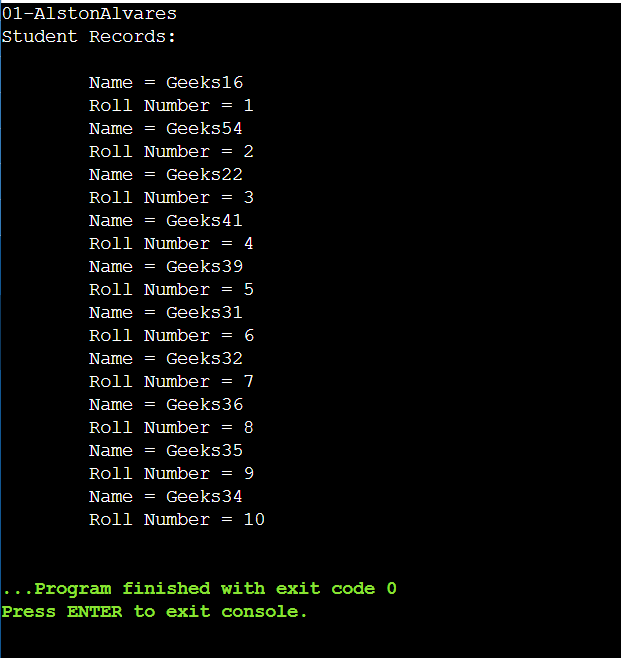
*printf("\tRoll Number = %d\n", student[i].roll\_number);*

*}*

*return 0;*

*}*

**Output:**



**Figure-20**

**Conclusion :** Successfully performed a program to print rollno and names of 10 students using array.

**Experiment no – 06(b)**

**Aim: Write a program to read a matrix of size m\*n.**

**Algorithm:**

1. Start
2. Enter row and column size
3. Construct Matrix
4. Display result
5. Stop

**Code:**

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares.\n");*

*int i,j,m,n;*

*float a[10][10];*

*printf("Enter row and column size:\n");*

*scanf("%d%d", &m, &n);*

*printf("Enter matrix elements:\n");*

*for(i=0;i< m;i++)*

*{*

*for(j=0;j< n;j++) {*

*printf("a[%d][%d]=",i,j);*

*scanf("%f", &a[i][j]);*

*}*

*}*

*printf("Matrix read is:\n");*

*for(i=0;i< m;i++)*

*{*

*for(j=0;j< n;j++)*

*{*

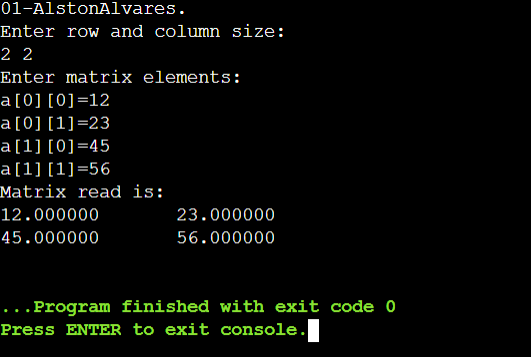
*printf("%f\t",a[i][j]);*

*}*

*printf("\n");*

*}}*

**Output:**



**Figure-21**

**Conclusion :** Successfully performed a program to read a matrix of size m\*n**.**

**Experiment no – 06(c)**

**Aim**: **Write a program to sort the elements of array in ascending or descending order.**

**Algorithm:**

1. Start.
2. Input size of array.
3. Place currently selected element array to its correct place.
4. Swap if currently selected array element to its correct place.
5. Print the sorted array.
6. Stop.

**Code:**

*/\*\**

*\* C program to sort elements of array in ascending order*

*\*/*

*#include <stdio.h>*

*#define MAX\_SIZE 100 // Maximum array size*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int arr[MAX\_SIZE];*

*int size;*

*int i, j, temp;*

*/\* Input size of array \*/*

*printf("Enter size of array: ");*

*scanf("%d", &size);*

*/\* Input elements in array \*/*

*printf("Enter elements in array: ");*

*for(i=0; i<size; i++)*

*{*

*scanf("%d", &arr[i]);*

*}*

*for(i=0; i<size; i++)*

*{*

*/\**

*\* Place currently selected element array[i]*

*\* to its correct place.*

*\*/*

*for(j=i+1; j<size; j++)*

*{*

*/\**

*\* Swap if currently selected array element*

*\* is not at its correct position.*

*\*/*

*if(arr[i] > arr[j])*

*{*

*temp = arr[i];*

*arr[i] = arr[j];*

*arr[j] = temp;*

*}*

*}*

*}*

*/\* Print the sorted array \*/*

*printf("\nElements of array in ascending order: ");*

*for(i=0; i<size; i++)*

*{*

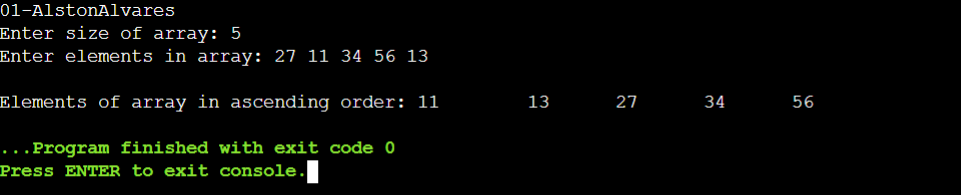
*printf("%d\t", arr[i]);*

*}*

*return 0;*

*}*

**Output:**



**Figure-22**

**Conclusion :** Successfully performed a program to sort the elements of array in ascending or descending order

**Experiment no – 07(a)**

**Aim**: **Write a program to extract the portion of a character string and print the extracted part.**

**Algorithm:**

1. Start
2. Enter index start
3. Enter index last
4. Print result
5. Stop

**Code:**

*#include<stdio.h>*

*#include<conio.h>*

*#include<string.h>*

*void demo(char\*s,int start,int end)*

*{*

*int i;*

*for(i=start;i<=end;i++)*

*printf("%c",s[i]);*

*}*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*char str[100]="NAVI MUMBAI";*

*int s,e;*

*printf("Enter the Start Index:");*

*scanf("%d",&s);*

*printf("Enter the Last Index:");*

*scanf("%d",&e);*

*if(e>strlen(str) ||(s>strlen(str)))*

*printf("The indeex's starting or ending value is out of range ");*

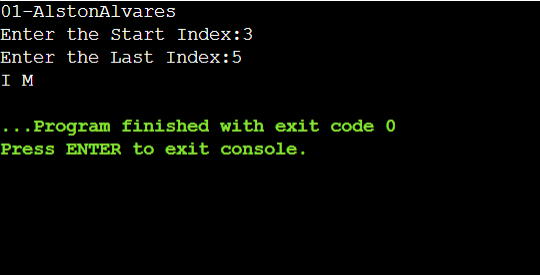
*else*

*demo(str,s,e);*

*return 0;*

*}*

**Output:**



**Figure-23**

**Conclusion :** Successfully performed a program to extract the portion of a character string and print the extracted part.

**Experiment no – 07(b)**

**Aim**: **Write a program to find the given string is palindrome or not.**

**Algorithm:**

1. Start
2. Check “hello” = palindrome
3. If true then print is a palindrome
4. If false then print is not a palindrome
5. Check “madam”= palindrome
6. If true then print is a palindrome
7. If false then print is not a palindrome
8. Stop.

**Code:**

**#***include<stdio.h>*

*#include<string.h>*

*void isPal(char s[])*

*{*

*int l = 0;*

*int h = strlen(s)-1;*

*while(h>l)*

*{*

*if(s[l++]!=s[h--])*

*{*

*printf("%s:not a palindrome\n",s);*

*return;*

*}*

*}*

*printf("%s :palindrome\n",s);*

*}*

*int main()*

*{ printf("01-AlstonAlvares\n");*

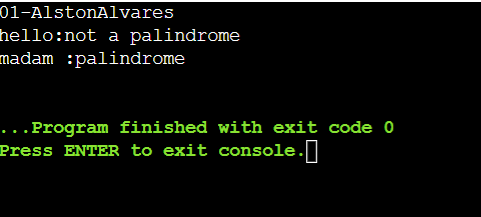
*isPal("hello");*

*isPal("madam");*

*return 0;*

*}*

**Output:**



**Figure-24**

**Conclusion :** Successfully performed a program to find the given string is palindrome or not.

**Experiment no – 07(c)**

**Aim**: **Write a program to using strlen(), strcmp() function.**

**Algorithm:**

1. Start
2. Use strlen function
3. Use strcmp function
4. Print result
5. Stop

**Code:**

*#include<stdio.h>*

*#include<string.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int i;*

*i=strlen("Hello");*

*printf("\n %d",i);*

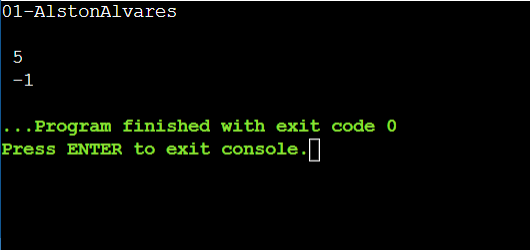
*i=strcmp("Hello!","World");*

*printf("\n %d",i);*

*return 0;*

*}*

**Output:**



**Figure-25**

**Conclusion :** Successfully performed a program to using strlen(), strcmp() function.

**Experiment no – 08(a)**

**Aim**: **Write a program to display the values using different data types and its address using pointer.**

**Algorithm:**

1. Start
2. Declare v1,v2,v3
3. Declare \*p1,\*p2,\*p3
4. Insert values
5. Print result
6. Stop

**Code:**

*#include <stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int v1;*

*float v2;*

*char v3;*

*int \*p1;*

*float \*p2;*

*char \*p3;*

*v1=11;*

*v2=3.14;*

*v3='Y';*

*p1 = &v1;*

*p2 = &v2;*

*p3 = &v3;*

*printf("Address of v1 = %u\n", &v1); printf("Value is = %d\n", \*p1);*

*printf("Address of v2 = %u\n", &v2);*

*printf("Value is = %f\n", \*p2);*

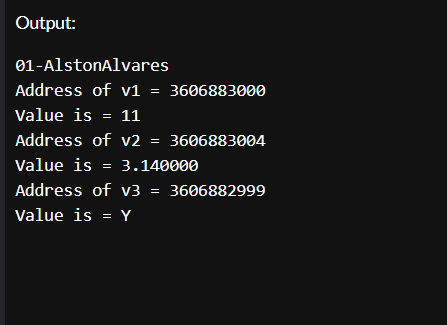
*printf("Address of v3 = %u\n", &v3);*

*printf("Value is = %c\n", \*p3);*

*return 0;*

*}*

**Output:**



**Figure-26**

**Conclusion :** Successfully performeda program to display the values using different data types and its address using pointer**.**

**Experiment no – 08(b)**

**Aim:** **Write a program to perform addition and subtraction using pointer.**

**Algorithm:**

1. Start
2. Enter numbers
3. Addition or Subtraction is performed
4. Display results
5. Stop

**Code:**

*//Add//*

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int num1 ,num2, \*p,\*q,sum;*

*printf("Enter any two integers:\n");*

*scanf("%d%d", &num1,&num2);*

*p = &num1;*

*q = &num2;*

*sum = \*p+\*q;*

*printf("Sum= %d\n",sum);*

*return 0;*

*}*

*//Sub//*

*#include<stdio.h>*

*int main()*

*{ printf("01-AlstonAlvares\n");*

*int num1 ,num2, \*p,\*q,sub;*

*printf("Enter any two integers:\n");*

*scanf("%d%d", &num1,&num2);*

*p = &num1;*

*q = &num2;*

*sub = \*p-\*q;*

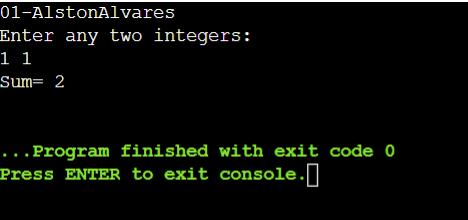
*printf("Sub= %d\n",sub);*

*return 0;*

*}*

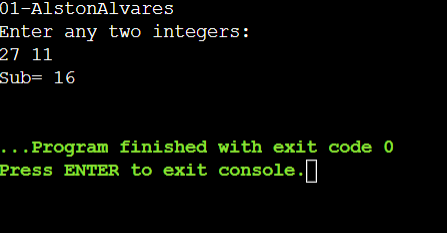
**Output:**

Add:



**Figure-27**

Sub:



**Figure-28**

**Conclusion :** Successfully performed a program to perform addition and subtraction using pointer.

**Experiment no – 09(a)**

**Aim**: **Write a program to copy the contents of the file from one file into other.**

**Algorithm:**

1. Start
2. Create 2 files, f1 and f2
3. Add text to f1
4. Use getc, putc, FILE.
5. Open f2
6. File f1’s content has been copied to f2
7. Stop

**Code:**

*#include<stdio.h>*

*main(){*

*FILE \*fp1, \*fp2;*

*char ch;*

*fp1 = fopen("f1.txt", "r");*

*fp2 = fopen("f2.txt", "w");*

*while((ch = getc(fp1)) != EOF)*

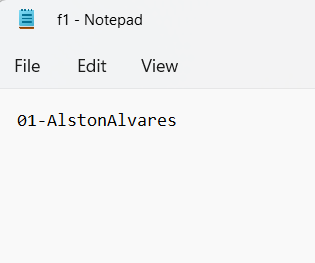
*putc(ch, fp2);*

*fclose(fp1);*

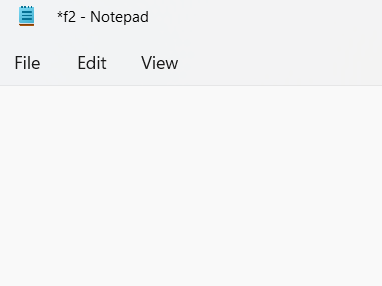
*fclose(fp2);*

*getch();*

*}*

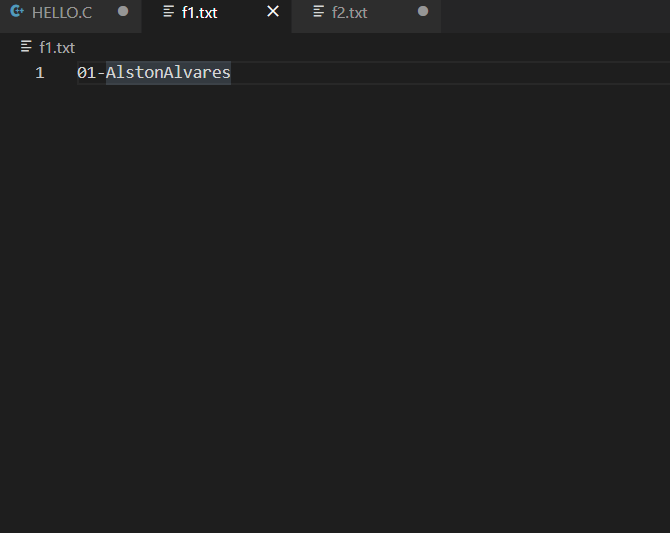
**

**Figure-29**

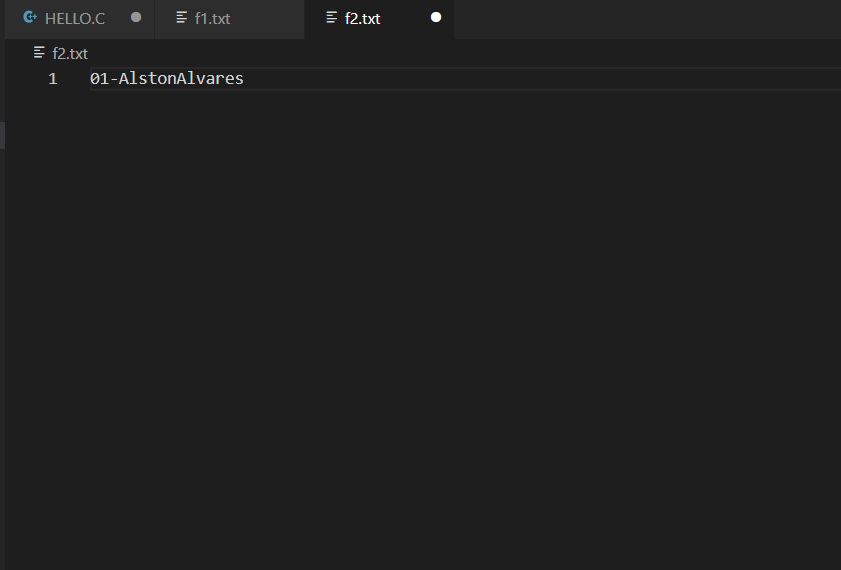
**

**Figure-30**

**Output:**



**Figure-31**



**Figure-32**

**Conclusion :** Successfully performed a program to copy the contents of the file from one file into other.

**Experiment no – 09(b)**

**Aim**: **Write a program to print the structure using**

**• Title**

**• Author**

**• Subject**

**• Book ID Print the details of two students.**

**Code:**

*#include<stdio.h>*

*struct book{*

*char Title[40];*

*char Author[40];*

*char Subject[40];*

*int Book\_ID;*

*};*

*int main() {*

*printf("01-AlstonAlvares\n.");*

*struct book b[3];*

*int i;*

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

*{*

*printf("Enter details of book #%d\n", i+1);*

*printf("Enter book Id: ");*

*scanf("%d", &b[i].Book\_ID);*

*printf("Enter book Title: ");*

*scanf("%s", &b[i].Title);*

*printf("Enter book Subject: ");*

*scanf("%s", &b[i].Subject);*

*printf("Enter book Author: ");*

*scanf("%s", &b[i].Author); }*

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

*{*

*printf("\nBook %d........ .\n\n", i+1);*

*printf("Book Id: %d\n", b[i].Book\_ID);*

*printf("Book Name: %s\n", b[i].Title);*

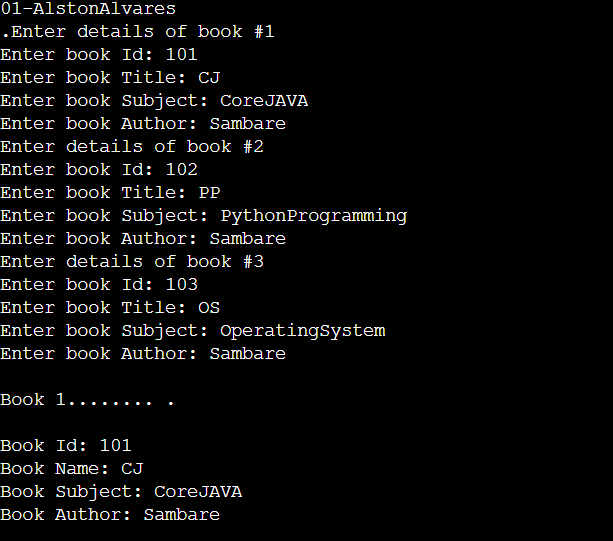
*printf("Book Subject: %s\n", b[i].Subject);*

*printf("Book Author: %s\n", b[i].Author); }*

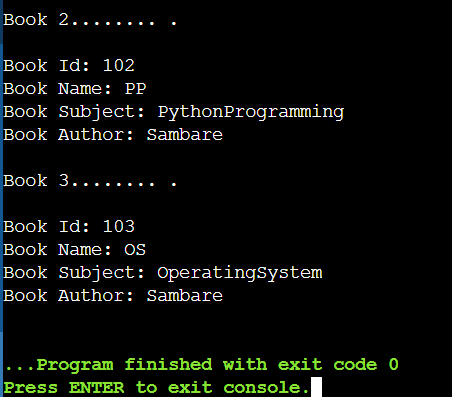
*return 0;*

*}*

**Output:**



**Figure-33**



**Figure-34**

**Conclusion :** Successfully performed a program to print the details of two students.

**Experiment no – 10**

**Aim**: **Create a mini project on “Bank management system”. The program should be menu driven**

**Algorithm:**

1. Start
2. Enter number of customers record to enter
3. Read the number
4. Enter account number
5. Enter name
6. Display Press I to deposit amount, Press 2 to withdraw amount, Press 0 to Exit.
7. Stop

**Code:**

*#include <stdio.h>*

*struct customer {*

*int account\_no;*

*char name[80];*

*int balance;*

*};*

*void accept(struct customer[], int);*

*int search(struct customer[], int, int);*

*void deposit(struct customer[], int, int, int);*

*void withdraw(struct customer[], int, int, int);*

*int main()*

*{*

*struct customer data[20];*

*int n, choice, account\_no, amount, index;*

*printf("Banking System\n\n");*

*printf("Number of customer records you want to enter? :");*

*scanf("%d", &n);*

*accept(data, n);*

*do {*

*printf("\nBanking System Menu:\n");*

*printf("Press I to deposit amount.\n");*

*printf("Press 2 to withdraw amount.\n");*

*printf("Press 0 to exit\n");*

*printf("\nEnter choice(0-4): ");*

*scanf("%d", &choice);*

*switch (choice)*

*{*

*case 1:*

*printf("Enter account number: ");*

*scanf("%d", &account\_no);*

*printf("Enter amount to deposit: ");*

*scanf("%d", &amount);*

*deposit(data, n, account\_no, amount);*

*break;*

*case 2:*

*printf("Enter account number: ");*

*scanf("%d",&account\_no),*

*printf("Enter amount to withdraw :");*

*scanf("%d",&amount);*

*withdraw(data, n, account\_no, amount);*

*}*

*}*

*while (choice != 0);*

*return 0;*

*}*

*void accept(struct customer list[80], int s)*

*{*

*int i;*

*for (i = 0; i < s; i++)*

*{*

*printf("\nEnter data for Record #%d", i + 1);*

*printf("\nEnter account\_no: ");*

*scanf("%d", &list[i].account\_no);*

*printf("01-AlstonAlvares ");*

*gets(list[i].name);*

*list[i].balance = 0;*

*} }*

*int search(struct customer list[80], int s, int number)*

*{*

*int i;*

*for (i = 0; i < s; i++)*

*{*

*if (list[i].account\_no == number) {*

*return i;*

*} }*

*return -1; }*

*void deposit(struct customer list[], int s, int number, int amt)*

*{*

*int i= search(list, s, number);*

*if (i == -1) {*

*printf("Record not found"); }*

*else{*

*list[i].balance+=amt;*

*} }*

*void withdraw(struct customer list[], int s, int number, int amt)*

*{*

*int i=search(list, s, number);*

*if(i== -1)*

*{*

*printf("Record not found\n");*

*}*

*else if (list[i].balance <amt) {*

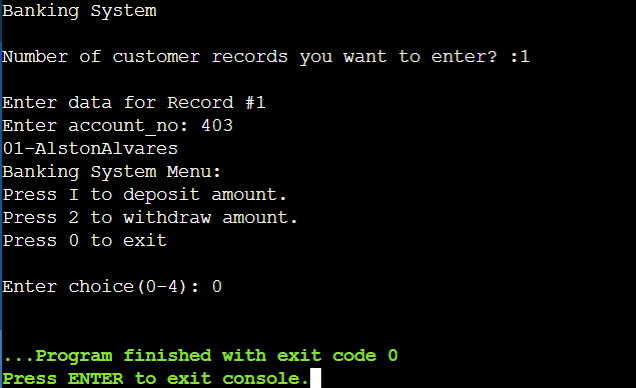
*printf("Insufficient balance\n"); }*

*Else {*

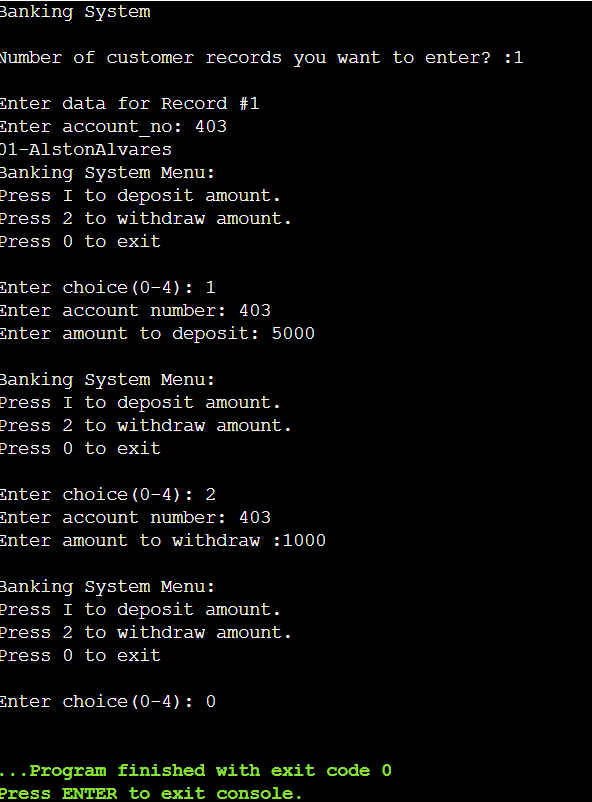
*list[i].balance-=amt;*

*} }*

**Output:**



**Figure-35**



**Figure-36**

**Conclusion :** Successfully performed a mini project on “Bank management system” . The program should be menu driven.