|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Programming in C** | **Semester:** | **II** |
| **Date of Performance:** | **24 / 01 / 2025** | **DIV/ Batch No:** | **C4-1** |
| **Student Name:** | **Dhruv Pankhania** | **Roll No:** | **16010124216** |

**Experiment No: 3**

**Title: Write a program in C to demonstrate use of looping control structures**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Write a menu-driven program for the following option  a.To find whether a number is palindrome or not. (e.g. 1221 is palindrome) using while loop  b. To calculate the sum of the Fibonacci series up to ‘n’ terms(use do-while loop only)  c. Write a program in C to make such a pattern as a right angle triangle with a number that will repeat a number in a row. |

|  |
| --- |
| **COs to be achieved:** |
| **CO:** Apply basic concepts of C programming for problem-solving.(CO1 and CO2). |

|  |
| --- |
| **Theory:** |
| Loops in programming are used to repeat a block of code until the specified condition is met. A loop statement allows programmers to execute a statement or group of statements multiple times without repetition of code.  There are mainly two types of loops in C Programming:   * Entry Controlled loops: In Entry controlled loops the test condition is checked before entering the main body of the loop. **For Loop and While Loop is Entry-controlled loops**. * Exit Controlled loops: In Exit controlled loops the test condition is evaluated at the end of the loop body. The loop body will execute at least once, irrespective of whether the condition is true or false. **do-while Loop is Exit Controlled loop**.   **for Loop**  for loop in C programming is a repetition control structure that allows programmers to write a loop that will be executed a specific number of times. for loop enables programmers to perform n number of steps together in a single line.  Syntax:  for (initialize expression; test expression; update expression)  {  //  // body of for loop  //  }  For Example:-  for(int i = 0; i < n; ++i)  {  printf("Body of for loop which will execute till n");  }  **While Loop**  While loop does not depend upon the number of iterations. In for loop the number of iterations was previously known to us but in the While loop, the execution is terminated on the basis of the test condition. If the test condition will become false then it will break from the while loop else body will be executed.  while (test\_expression)  {  // body of the while loop    update\_expression;  }  **do-while Loop**  The do-while loop is similar to a while loop but the only difference lies in the do-while loop test condition which is tested at the end of the body. In the do-while loop, the loop body will execute at least once irrespective of the test condition.  do  {  // body of do-while loop    update\_expression;  } while (test\_expression); |

|  |
| --- |
| **Problem Statements:** |
| 1. To find whether a number is palindrome or not. (e.g. 1221 is palindrome) using while loop 2. To calculate the sum of the Fibonacci series up to ‘n’ terms(use do-while loop only) 3. Write a program in C to make such a pattern like a right angle triangle with a number which will repeat a number in a row or as pattern given below   \*  \* \* \*  \* \* \* \* \*  \* \* \* \* \* \* \* |

|  |
| --- |
| **Code :** |
| 1. Check Palindrome  #include<stdio.h>  *int* main(){  *int* num, revNum, key;  *int* digit = 0;    printf("Enter a Number: ");    scanf("%d", &num);    key = num;  *int* digitCount = 0;    while(num){      num = num/10;      digitCount++;    }    num = key;    while(digitCount>=0){      digit += num%10;      for(*int* i = digitCount-1; i>0; i--){      digit \*= 10;      }      revNum +=digit;      digit=0;      num /= 10;      digitCount--;    }    if(key==revNum){      printf("%d is a Palindrom Number", key);    }    else{      printf("%d is not a Palindrom Number", key);    }      return 0;  }  2. Sum of Fibonacci Series  #include<stdio.h>  *int* main(){  *int* n, sum, a = 0, i = 1, j = 1;    printf("Enter a Number: ");    scanf("%d", &n);  do  {    sum += i;  *int* temp = i;    i = j;    j = temp + j;    a++;  }  while(a<n);  printf("\nSum of first %d is %d", n, sum);  return 0;  }  3. Print Pyramid using asterisk  #include<stdio.h>  *int* main(){  *int* a;    printf("Enter a Number: ");    scanf("%d", &a);      for(*int* i = 0; i < a; i++){      for(*int* j = 0; j < a - i - 1; j++){        printf(" ");      }      for(*int* k = 0; k < (2 \* i + 1); k++){        printf("\*");      }      printf("\n");    }      return 0;  } |

|  |
| --- |
| **Output:** |
| 1.  2.  3. |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. Write a program in C to display the n terms of a harmonic series and their sum. 1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms #include<stdio.h>  *int* main(){  *int* a;  *double* sum = 0.0;    printf("Enter a Number: ");    scanf("%d", &a);    for(*int* i=1; i<=a; i++){      printf("1/%d", i);      if(i!=a){        printf(" + ");      }      sum += 1.0 / i;    }    printf("\nSum of the harmonic series: %.6f\n", sum);    return 0;  }   1. Write a C program that displays the n terms of square natural numbers and their sum. 1 4 9 16 ... n Terms  #include<stdio.h>   *int* main(){  *int* n, sum=0;    printf("Enter a Number: ");    scanf("%d", &n);    for(*int* i =1; i<=n; i++){      printf("%d \t", i\*i);      sum += i\*i;    }    printf("\nSum of first %d natural numbers is %d", n, sum);    return 0;  } |

|  |
| --- |
| **Conclusion:** |
| We learned about three fundamental looping constructs: for loop, while loop, and do-while loop, each serving different purposes in iteration. The for loop is used when the number of iterations is known beforehand, consisting of an initialization, condition, and increment/decrement. The while loop executes as long as a given condition remains true, making it useful when the number of iterations is uncertain. Unlike the while loop, the do-while loop ensures the loop body executes at least once before checking the condition, making it suitable for scenarios where execution must happen at least once regardless of the condition. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |