**Module 2:**

1. **Printing Patterns Using Loops**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main()

{

int n;

scanf("%d", &n);

int len = 2\*n - 1;

for (int i = 0; i < len; i++) {

for (int j = 0; j < len; j++) {

int min = i < j ? i : j;

min = min < len-i ? min : len-i-1;

min = min < len-j-1 ? min : len-j-1;

printf("%d ", n-min);

}

printf("\n");

}

return 0;

}

Output

**A screenshot of a computer

Description automatically generated**

1. Correctness and loop invariants

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <assert.h>

void insertionSort(int N, int arr[]) {

    int i,j;

    int value;

    for(i=1;i<N;i++)

    {

        value=arr[i];

        j=i-1;

        while(j>=0 && value<arr[j])

        {

            arr[j+1]=arr[j];

            j=j-1;

            }

        arr[j+1]=value;

    }

    for(j=0;j<N;j++)

    {

        printf("%d",arr[j]);

        printf(" ");

}

}

int main(void) {

int N;

scanf("%d", &N);

int arr[N], i;

for(i = 0; i < N; i++) {

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

}

insertionSort(N, arr);

return 0;

}

**A black screen with white text

Description automatically generated**

1. Small And Large Triangle

#include <stdio.h>

// Function to calculate the area of a triangle

float calculateArea(float base, float height) {

return (0.5 \* base \* height);

}

int main() {

float base1, height1, base2, height2, area1, area2;

// Input for the first triangle

printf("Enter the base and height of the first triangle: ");

scanf("%f %f", &base1, &height1);

// Input for the second triangle

printf("Enter the base and height of the second triangle: ");

scanf("%f %f", &base2, &height2);

// Calculate the areas of both triangles

area1 = calculateArea(base1, height1);

area2 = calculateArea(base2, height2);

// Compare the areas and determine which triangle is larger

printf("\nArea of first triangle: %.2f", area1);

printf("\nArea of second triangle: %.2f", area2);

if (area1 > area2) {

printf("\nThe first triangle is larger.\n");

printf("The second triangle is smaller.\n");

} else if (area1 < area2) {

printf("\nThe second triangle is larger.\n");

printf("The first triangle is smaller.\n");

} else {

printf("\nBoth triangles have the same area.\n");

}

return 0;

}

**A screenshot of a computer

Description automatically generated**

1. Happy Number

#include <stdio.h>

int getSumOfSquares(int num)

{

int sum = 0;

while (num > 0)

{

int digit = num % 10;

sum += digit \* digit; num /= 10;

}

return sum;

}

int isHappy(int num)

{

int slow = num, fast = num;

do

{

slow = getSumOfSquares(slow); // Move slow pointer by one step

fast = getSumOfSquares(getSumOfSquares(fast)); // Move fast pointer by two steps

} while (slow != fast);

return (slow == 1); // If they meet at 1, it's a happy number

}

int main()

{

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

if (isHappy(num))

{

printf("%d is a happy number.\n", num);

}

Else

{

printf("%d is not a happy number.\n", num);

}

return 0;

}

1. Triangle Numbers

#include <stdio.h>

// Function to calculate the nth triangular number

int triangularNumber(int n) {

return (n \* (n + 1)) / 2;

}

int main() {

int n, i;

// Input for how many triangular numbers to generate

printf("Enter the number of triangular numbers to generate: ");

scanf("%d", &n);

// Generate and display the first n triangular numbers

printf("The first %d triangular numbers are:\n", n);

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

printf("%d ", triangularNumber(i));

}

printf("\n");

return 0;

}

A screenshot of a computer

Description automatically generated

**Module 3:**

1. **For Loop in C**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main()

{

int a, b,i;

scanf("%d\n%d", &a, &b);

// Complete the code.

for(i=a;i<=b;i++)

{

if(i<10)

{

if(i==1)

printf("one\n");

else if(i==2)

printf("two\n");

else if(i==3)

printf("three\n");

else if(i==4)

printf("four\n");

else if(i==5)

printf("five\n");

else if(i==6)

printf("six\n");

else if(i==7)

printf("seven\n");

else if(i==8)

printf("eight\n");

else if(i==9)

printf("nine\n");

}

else {

if(i%2==1)

printf("odd\n");

else {

printf("even\n");

}

}

}

return 0;

}

Output:

A screenshot of a computer

Description automatically generated

1. **Calculate the Nth term**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

//Complete the following function.

int find\_nth\_term(int n, int a, int b, int c) {

//Write your code here.

int term, t1 = a, t2 = b, t3 = c;

if (n == 1)

term = t1;

else if (n == 2)

term = t2;

else if (n == 3)

term = t3;

else {

for (int i = 4; i <= n; i++) {

term = t1 + t2 + t3;

t1 = t2;

t2 = t3;

t3 = term;

}

}

return term;

}

int main() {

int n, a, b, c;

scanf("%d %d %d %d", &n, &a, &b, &c);

int ans = find\_nth\_term(n, a, b, c);

printf("%d", ans);

return 0;

}

Output:

A screenshot of a computer

Description automatically generated

1. **Students Marks Sum**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

//Complete the following function.

int marks\_summation(int\* marks, int number\_of\_students, char gender) {

//Write your code here.

int sum = 0;

for(int i = (gender == 'b' ? 0 : gender == 'g' ? 1 : -1); i < number\_of\_students; i+=2) {

sum += marks[i];

}

return sum;

}

int main() {

int number\_of\_students;

char gender;

int sum;

scanf("%d", &number\_of\_students);

int \*marks = (int \*) malloc(number\_of\_students \* sizeof (int));

for (int student = 0; student < number\_of\_students; student++) {

scanf("%d", (marks + student));

}

scanf(" %c", &gender);

sum = marks\_summation(marks, number\_of\_students, gender);

printf("%d", sum);

free(marks);

return 0;

}

Output:

A screenshot of a computer

Description automatically generated

1. Variadic Functions

#include <stdio.h>

#include <stdarg.h>

// Variadic function to calculate the sum of given integers

int sum(int count, ...) {

va\_list args;

int total = 0;

// Initialize the argument list

va\_start(args, count);

// Loop through all the arguments

for (int i = 0; i < count; i++) {

total += va\_arg(args, int); // Retrieve the next argument

}

// Clean up the argument list

va\_end(args);

return total;

}

int main() {

// Example usage of the sum function

int result1 = sum(3, 10, 20, 30); // Sum of 3 numbers: 10, 20, 30

int result2 = sum(5, 1, 2, 3, 4, 5); // Sum of 5 numbers: 1, 2, 3, 4, 5

// Display the results

printf("The sum of 10, 20, 30 is: %d\n", result1);

printf("The sum of 1, 2, 3, 4, 5 is: %d\n", result2);

return 0;

}

**A black and white screen with white text

Description automatically generated**

1. Nth Tribonacci number

**#include <stdio.h>**

**// Function to calculate the Nth Tribonacci number**

int tribonacci(int n) {

if (n == 0) return 0;

if (n == 1 || n == 2) return 1;

int a = 0, b = 1, c = 1, next;

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

next = a + b + c; // Calculate the next term

a = b; // Update a to the next term

b = c; // Update b to the next term

c = next; // Update c to the next term

}

return c;

}

int main() {

int n;

// Input the value of N

printf("Enter the value of N: ");

scanf("%d", &n);

// Calculate and display the Nth Tribonacci number

printf("The %dth Tribonacci number is: %d\n", n, tribonacci(n));

return 0;

}

**A screenshot of a computer

Description automatically generated**