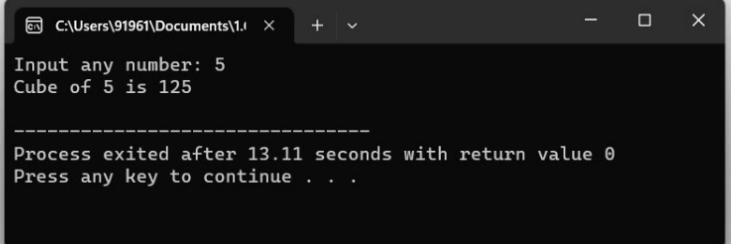


1.Cube of network classwork 3.cpp

```
1 #include <stdio.h>
2
3 // function to find cube of a given number
4 int cube(int num) {
5     return num * num * num;
6 }
7
8 int main() {
9     int num;
10    printf("Input any number: ");
11    scanf("%d", &num);
12
13    // calling the cube function to find the cube of the given number
14    int cube_num = cube(num);
15
16    printf("Cube of %d is %d\n", num, cube_num);
17    return 0;
18 }
```

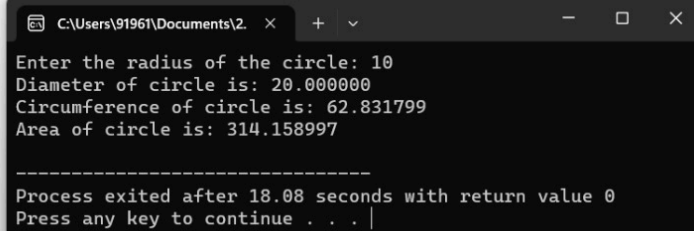


```
C:\Users\91961\Documents\1.1
Input any number: 5
Cube of 5 is 125

-----
Process exited after 13.11 seconds with return value 0
Press any key to continue . . .
```

2. calculate circular properties classwork 3.cpp

```
1 #include <stdio.h>
2
3 // Function to calculate diameter of circle
4 float diameter(float radius) {
5     return 2 * radius;
6 }
7
8 // Function to calculate circumference of circle
9 float circumference(float radius) {
10     return 2 * 3.14159 * radius;
11 }
12
13 // Function to calculate area of circle
14 float area(float radius) {
15     return 3.14159 * radius * radius;
16 }
17
18 int main() {
19     float radius;
20
21     printf("Enter the radius of the circle: ");
22     scanf("%f", &radius);
23
24     printf("Diameter of circle is: %f\n", diameter(radius));
25     printf("Circumference of circle is: %f\n", circumference(radius));
26     printf("Area of circle is: %f\n", area(radius));
27
28     return 0;
29 }
```

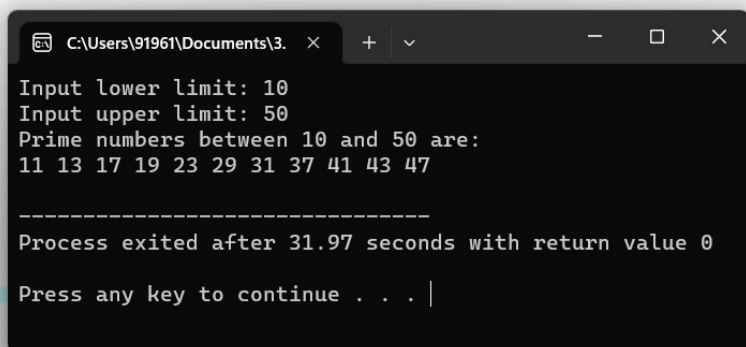


```
C:\Users\91961\Documents\2. x + v
Enter the radius of the circle: 10
Diameter of circle is: 20.000000
Circumference of circle is: 62.831799
Area of circle is: 314.158997

-----
Process exited after 18.08 seconds with return value 0
Press any key to continue . . . |
```

3.PRIME.cpp

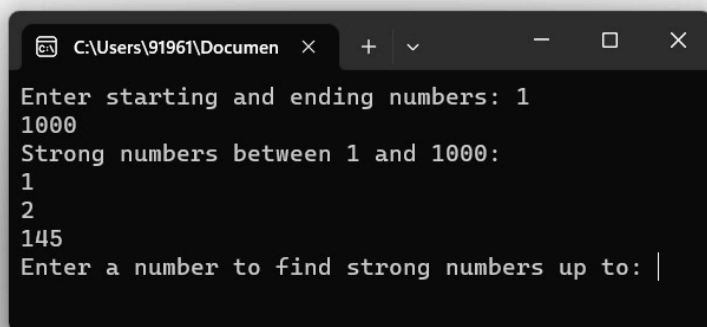
```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 bool is_prime(int num) {
5     if (num < 2) {
6         return false;
7     }
8     for (int i = 2; i <= num/2; i++) {
9         if (num % i == 0) {
10             return false;
11         }
12     }
13     return true;
14 }
15
16 void display_primes(int lower, int upper) {
17     printf("Prime numbers between %d and %d are:\n", lower, upper);
18     for (int i = lower; i <= upper; i++) {
19         if (is_prime(i)) {
20             printf("%d ", i);
21         }
22     }
23     printf("\n");
24 }
25
26 int main() {
27     int lower, upper;
28
29     printf("Input lower limit: ");
30     scanf("%d", &lower);
31     printf("Input upper limit: ");
32     scanf("%d", &upper);
33
34     display_primes(lower, upper);
35
36     return 0;
37 }
```



```
C:\Users\91961\Documents\3. x + v - □ ×
Input lower limit: 10
Input upper limit: 50
Prime numbers between 10 and 50 are:
11 13 17 19 23 29 31 37 41 43 47
-----
Process exited after 31.97 seconds with return value 0
Press any key to continue . . . |
```

4.STRONG.cpp

```
11 // Function to check if a number is a strong number
12 int isStrong(int n) {
13     int sum = 0, temp = n;
14     while (temp != 0) {
15         sum += factorial(temp % 10);
16         temp /= 10;
17     }
18     return (sum == n);
19 }
20
21 // Function to print all strong numbers in a given range
22 void printStrong(int start, int end) {
23     printf("Strong numbers between %d and %d:\n", start, end);
24     for (int i = start; i <= end; i++) {
25         if (isStrong(i)) {
26             printf("%d\n", i);
27         }
28     }
29 }
30
31 // Function to find all strong numbers between 1 and n
32 void findStrong(int n) {
33     printf("Strong numbers between 1 and %d:\n", n);
34     for (int i = 1; i <= n; i++) {
35         if (isStrong(i)) {
36             printf("%d\n", i);
37         }
38     }
39 }
40
41 // Main function
42 int main() {
43     int start, end, n;
44     printf("Enter starting and ending numbers: ");
45     scanf("%d %d", &start, &end);
46     printStrong(start, end);
47     printf("Enter a number to find strong numbers up to: ");
48     scanf("%d", &n);
49     findStrong(n);
50     return 0;
51 }
52
```



```
C:\Users\91961\Documen x + - □ ×
Enter starting and ending numbers: 1
1000
Strong numbers between 1 and 1000:
1
2
145
Enter a number to find strong numbers up to: |
```

```

5.function prototype.cpp
1  #include <stdio.h>
2
3  int isEven(int n); // Function prototype
4
5  int main() {
6      int num;
7      printf("Enter a number: ");
8      scanf("%d", &num);
9
10     if (isEven(num)) {
11         printf("%d is even.\n", num);
12     } else {
13         printf("%d is odd.\n", num);
14     }
15
16     return 0;
17 }
18
19 int isEven(int n) {
20     return (n % 2 == 0);
21 }

```

```

C:\Users\91961\Documents\5.
Enter a number: 10
10 is even.

-----
Process exited after 11.09 seconds with return value 0
Press any key to continue . . .

```

6.Armstrong.cpp

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int isArmstrong(int num);
5
6 void printArmstrongNumbers(int start, int end) {
7     int i;
8     printf("Armstrong numbers between %d and %d are:\n", start, end);
9
10    for(i = start; i <= end; i++) {
11        if(isArmstrong(i)) {
12            printf("%d ", i);
13        }
14    }
15 }
16
17
18 int isArmstrong(int num) {
19     int sum = 0, rem, temp;
20     int digits = 0;
21
22     temp = num;
23
24     while(temp != 0) {
25         digits++;
26         temp /= 10;
27     }
28
29     temp = num;
30
31     while(temp != 0) {
32         rem = temp % 10;
33         sum += pow(rem, digits);
34         temp /= 10;
35     }
36
37     return (num == sum);
38 }
39
40 int main() {
41     int start, end;
42
43     printf("Enter the starting and ending numbers: ");
44     scanf("%d %d", &start, &end);
45
46     printArmstrongNumbers(start, end);
47
48     return 0;
49 }
```

```
C:\Users\91961\Documents\6 x + - □ ×
Enter the starting and ending numbers: 1
1000
Armstrong numbers between 1 and 1000 are:
1 2 3 4 5 6 7 8 9 153 370 371 407
-----
Process exited after 109.4 seconds with return v
alue 0
Press any key to continue . . . |
```

7.function point of all perfect numbers.cpp

```
1  #include <stdio.h>
2
3  int isPerfect(int num) {
4      int sum = 0;
5      for (int i = 1; i < num; i++) {
6          if (num % i == 0) {
7              sum += i;
8          }
9      }
10     if (sum == num) {
11         return 1;
12     } else {
13         return 0;
14     }
15 }
16
17 void printPerfectNumbers(int start, int end) {
18     for (int i = start; i <= end; i++) {
19         if (isPerfect(i)) {
20             printf("%d ", i);
21         }
22     }
23 }
24
25 int main() {
26     int n;
27     printf("Enter the value of n: ");
28     scanf("%d", &n);
29     printf("Perfect numbers between 1 and %d are: ", n);
30     printPerfectNumbers(1, n);
31     return 0;
32 }
```

C:\Users\91961\Documents\7. × + ▾

Enter the value of n: 100

Perfect numbers between 1 and 100 are: 6 28

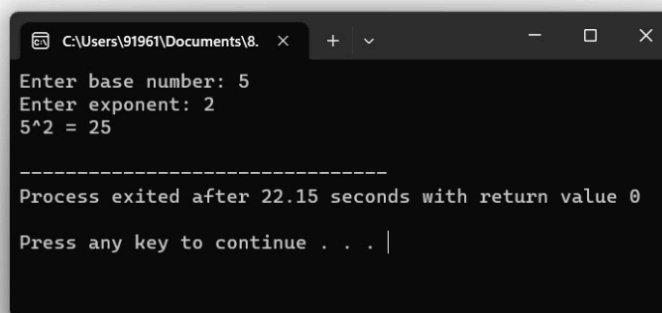
Process exited after 15.57 seconds with return value 0

Press any key to continue . . . |

```

8.input.cpp
1  #include <stdio.h>
2
3  int power(int base, int exponent);
4
5  int main() {
6      int base, exponent, result;
7      printf("Enter base number: ");
8      scanf("%d", &base);
9      printf("Enter exponent: ");
10     scanf("%d", &exponent);
11
12     result = power(base, exponent);
13
14     printf("%d^%d = %d\n", base, exponent, result);
15     return 0;
16 }
17
18 int power(int base, int exponent) {
19     if (exponent == 0) {
20         return 1;
21     } else if (exponent % 2 == 0) {
22         int temp = power(base, exponent / 2);
23         return temp * temp;
24     } else {
25         int temp = power(base, (exponent - 1) / 2);
26         return base * temp * temp;
27     }
28 }

```



```

C:\Users\91961\Documents\8.  x + - □ x
Enter base number: 5
Enter exponent: 2
5^2 = 25

-----
Process exited after 22.15 seconds with return value 0
Press any key to continue . . . |

```

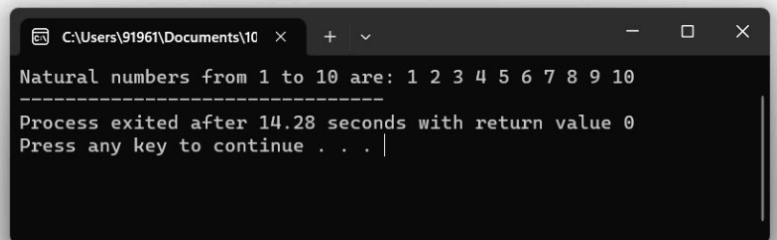

9.recursie function.cpp

```
1  #include <stdio.h>
2
3  void printNaturalNumbers(int n)
4  {
5      if(n==0) // base case
6          return;
7      else
8      {
9          printNaturalNumbers(n-1); // recursive call
10         printf("%d ", n); // print the current natural number
11     }
12 }
13
14 int main()
15 {
16     int n;
17     printf("Enter a positive integer: ");
18     scanf("%d", &n);
19     printf("Natural numbers from 1 to %d are: ", n);
20     printNaturalNumbers(n);
21     return 0;
22 }
```

```
C:\Users\91961\Documents\9.  x  +  v  -  □  ×
Enter a positive integer: 10
Natural numbers from 1 to 10 are: 1 2 3 4 5 6 7 8 9 10
-----
Process exited after 11.91 seconds with return value 0
Press any key to continue . . . |
```

10.logic to point.cpp

```
1  #include <stdio.h>
2
3  void printNaturalNumbers(int n)
4  {
5      if(n==0) // base case
6          return;
7      else
8      {
9          printNaturalNumbers(n-1); // recursive call
10         printf("%d ", n); // print the current natural number
11     }
12 }
13
14 int main()
15 {
16     int n;
17     printf("Enter a positive integer: ");
18     scanf("%d", &n);
19     printf("Natural numbers from 1 to %d are: ", n);
20     printNaturalNumbers(n);
21     return 0;
22 }
```



C:\Users\91961\Documents\10 x + v - □ ×

Natural numbers from 1 to 10 are: 1 2 3 4 5 6 7 8 9 10

Process exited after 14.28 seconds with return value 0
Press any key to continue . . . |

11.all natural numbers.cpp

```
1  #include<stdio.h>
2
3  int findSum(int n)
4  {
5      if (n == 0)
6          return 0;
7      else
8          return n + findSum(n-1);
9  }
10
11 int main()
12 {
13     int n;
14     printf("Enter the value of n: ");
15     scanf("%d", &n);
16     printf("The sum of natural numbers between 1 to %d is %d", n, findSum(n));
17     return 0;
18 }
```

```
C:\Users\91961\Documents\11  x  +  v  -  □  x
Enter the value of n: 10
The sum of natural numbers between 1 to 10 is 55
-----
Process exited after 19.08 seconds with return value 0
Press any key to continue . . . |
```

```

12.sum even.cpp
1 #include <stdio.h>
2
3 int sumEvenOdd(int start, int end, int type){
4     if (start > end){
5         return 0;
6     }
7     if(type == 0){ // for even numbers
8         if(start % 2 == 0){
9             return start + sumEvenOdd(start+2, end, 0);
10        }
11        else{
12            return sumEvenOdd(start+1, end, 0);
13        }
14    }
15    else{ // for odd numbers
16        if(start % 2 != 0){
17            return start + sumEvenOdd(start+2, end, 1);
18        }
19        else{
20            return sumEvenOdd(start+1, end, 1);
21        }
22    }
23 }
24
25 int main(){
26     int n, sum;
27     printf("Enter the value of n: ");
28     scanf("%d", &n);
29     sum = sumEvenOdd(1, n, 0); // 0 for even numbers, 1 for odd numbers
30     printf("Sum of all even numbers between 1 to %d is %d", n, sum);
31     return 0;
32 }

```

C:\Users\91961\Documents\12

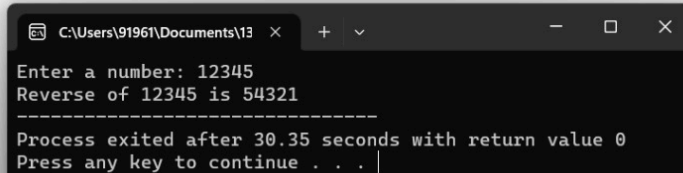
Enter the value of n: 100
Sum of all even numbers between 1 to 100 is 2550

Process exited after 11 seconds with return value 0
Press any key to continue . . .

Open a new tab
Alt+Click to split the current window
Shift+Click to open a new window

13.base case.cpp

```
1 #include <stdio.h>
2
3 int reverse(int num, int rev) {
4     // Base case: If the given number becomes zero, return the reversed number
5     if (num == 0) {
6         return rev;
7     } else {
8         // Recursive case: Calculate the last digit and add it to the reversed number
9         int last_digit = num % 10;
10        rev = rev * 10 + last_digit;
11        // Divide the number by 10 and recursively call the function
12        return reverse(num / 10, rev);
13    }
14 }
15
16 int main() {
17     int num;
18     printf("Enter a number: ");
19     scanf("%d", &num);
20     int rev_num = reverse(num, 0);
21     printf("Reverse of %d is %d", num, rev_num);
22     return 0;
23 }
```



C:\Users\91961\Documents\13 x + v - □ ×

Enter a number: 12345
Reverse of 12345 is 54321

Process exited after 30.35 seconds with return value 0
Press any key to continue . . . |

```

14.palindrom.cpp
1  #include <stdio.h>
2
3  int isPalindrome(int num, int temp) {
4      if (num < 10) {
5          return (temp == num);
6      }
7      if (num % 10 == temp % 10) {
8          return isPalindrome(num / 10, temp / 10);
9      }
10     return 0;
11 }
12
13 int main() {
14     int num;
15     printf("Enter a number: ");
16     scanf("%d", &num);
17     if (isPalindrome(num, num)) {
18         printf("%d is a palindrome number\n", num);
19     } else {
20         printf("%d is not a palindrome number\n", num);
21     }
22     return 0;
23 }

```

```

C:\Users\91961\Documents\14
Enter a number: 121
121 is a palindrome number

-----
Process exited after 23.17 seconds with return value 0
Press any key to continue . . .

```

```
15.sum of digits.cpp ×
1  #include <stdio.h>
2
3  int sumOfDigits(int num);
4
5  int main() {
6      int num, sum;
7      printf("Enter a number: ");
8      scanf("%d", &num);
9      sum = sumOfDigits(num);
10     printf("The sum of digits of %d is %d.\n", num, sum);
11     return 0;
12 }
13
14 int sumOfDigits(int num) {
15     if (num == 0) { // base case
16         return 0;
17     } else {
18         return (num % 10) + sumOfDigits(num / 10);
19     }
20 }
```

```
C:\Users\91961\Documents\15 × + - □ ×
Enter a number: 1234
The sum of digits of 1234 is 10.

-----
Process exited after 14.12 seconds with return value 0
Press any key to continue . . .
```

```

16.swap.cpp
1  #include <stdio.h>
2
3  void swap(int *a, int *b);
4
5  int main()
6  {
7      int num1, num2;
8
9      printf("Enter two integers: ");
10     scanf("%d %d", &num1, &num2);
11
12     printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
13
14     swap(&num1, &num2);
15
16     printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
17
18     return 0;
19 }
20
21 void swap(int *a, int *b)
22 {
23     int temp;
24
25     temp = *a;
26     *a = *b;
27     *b = temp;
28 }

```

```

C:\Users\91961\Documents\16
Enter two integers: 10
20
Before swapping: num1 = 10, num2 = 20
After swapping: num1 = 20, num2 = 10

-----
Process exited after 25.16 seconds with return value 0
Press any key to continue . . .

```



```

1  #include <stdio.h>
2
3  int main() {
4      int size;
5      printf("Input array size: ");
6      scanf("%d", &size);
7
8      int arr[size];
9      int *ptr = arr; // pointer to the first element of the array
10
11     printf("Input elements: \n");
12     for (int i = 0; i < size; i++) {
13         scanf("%d", ptr+i); // inputting elements using pointer
14     }
15
16     printf("Array elements: ");
17     for (int i = 0; i < size; i++) {
18         printf("%d ", *(ptr+i)); // printing elements using pointer
19     }
20     printf("\n");
21     return 0;
22 }
23

```

```

C:\Users\91961\Documents\17 x + - □ ×
Input array size: 10
Input elements:
1
2
3
4
5
6
7
8
9
10
Array elements: 1 2 3 4 5 6 7 8 9 10
-----
Process exited after 47.54 seconds with return value 0
Press any key to continue . . . |

```

18.size source.cpp

```
1 #include <stdio.h>
2
3 #define MAX_SIZE 10
4
5 void copy_array(int *source, int *destination, int size);
6
7 int main() {
8     int array1[MAX_SIZE], array2[MAX_SIZE];
9     int size, i;
10
11     // Read in the size of the array
12     printf("Enter the size of the array: ");
13     scanf("%d", &size);
14
15     // Read in the elements of array1
16     printf("Enter array1 elements: ");
17     for (i = 0; i < size; i++) {
18         scanf("%d", &array1[i]);
19     }
20
21     // Copy the elements of array1 to array2 using pointers
22     copy_array(array1, array2, size);
23
24     // Print out the elements of array2
25     printf("array2 elements: ");
26     for (i = 0; i < size; i++) {
27         printf("%d ", array2[i]);
28     }
29     printf("\n");
30
31     return 0;
32 }
33
34 void copy_array(int *source, int *destination, int size) {
35     int i;
36     for (i = 0; i < size; i++) {
37         *(destination + i) = *(source + i);
38     }
39 }
```

```
C:\Users\91961\Documents\18
Enter the size of the array: 5
Enter array1 elements: 20
30
40
50
60
array2 elements: 20 30 40 50 60

-----
Process exited after 24.18 seconds with return value 0
Press any key to continue . . .
```

```
19.void.cpp
1 #include <stdio.h>
2
3 void input(int *arr, int m, int n)
4 {
5     int i, j;
6     for(i=0;i<m;i++)
7     {
8         for(j=0;j<n;j++)
9         {
10             scanf("%d", *(arr+i)+j);
11         }
12     }
13 }
14
15 void print(int *arr, int m, int n)
16 {
17     int i, j;
18     for(i=0;i<m;i++)
19     {
20         for(j=0;j<n;j++)
21         {
22             printf("%d ", (*(arr+i)+j));
23         }
24         printf("\n");
25     }
26 }
27
28 int main()
29 {
30     int arr[3][3];
31     printf("Input elements in 3x3 matrix: \n");
32     input(&arr[0][0], 3, 3);
33     printf("Elements in the matrix are: \n");
34     print(&arr[0][0], 3, 3);
35     return 0;
36 }
```

Input elements in 3x3 matrix:
123

Process exited after 13.12 seconds with return value 322122547
7
Press any key to continue . . . |

Resources Compile Log Debug Find Results Console Close