1. Algorithm to Check Prime Numbers in C++

Run a loop from 2 to n/2 using a variable i.

Inside the for loop, check whether n is divisible by i using the modulo operator (n % i == 0).

If n is not divisible by i, it means n is a prime number. In other words, if n != 0.

If n is divisible by i, it means n has divisors other than 1 and itself, and thus, it is not a prime number.

1. Design a program using a flow chart to prompt the user to enter a score, then return a grade based on the following criteria:

|  |  |
| --- | --- |
| **Score** | **Grade** |
| 0-49 | E |
| 50-59 | D |
| 60-69 | C |
| 70-79 | B |
| 80-100 | A |

Int scores;

String grade;

Cout<<”enter the score: “;

Cin>>score;

If (score >= 80 && score <= 100)

{

Cout<<”A”;

}

Else if(score >= 70 && score <= 79)

{

Cout<<”B”;

}

Else if(score >= 60 && score <= 69)

{

Cout<<”C”;

}

Else if(score >= 50 && score <= 59)

{

Cout<<”D”;

}

Else

{

Cout<<”E”;

}

Algorithm

1. Read scores, grade
2. Input the score value
3. If score >= 80 && score <= 100
4. Print grade A
5. Else If score >= 70 && score <= 79
6. Print grade B
7. Else If score >= 60 && score <= 69
8. Print grade C
9. Else If score >= 50 && score <= 59
10. Print grade D
11. Else
12. Print grade E
13. Write a C program to prompt the user to enter his user name and password then validate the value and return the appropriate message using the criteria below:

|  |  |
| --- | --- |
| **Username and password** | **Remark** |
| User name=”program” password=”bbit” | Welcome, login successful |
| User name not “program ” or password not “bbit” | Access denied |

String username, password;

Cout<<”Enter the username: ”;

Cin>>username;

Cout<<”Enter the password: ”;

Cin>>password;

If(username == “program” && password == “bbit”)

{

Cout<<”Welcome, login successfull”;

}

Else

{

Cout<<”Access denied”;

}

1. TWIGA Sacco allows its customers to make deposits in a savings account. The amount deposited earns a 12% annual interest. Given that P is deposit, A is the amount accumulated in n years and r is the interest rate. Write a C program that prompts the user to enter the initial P and n in years, then computes A given that A=P x (1+r/100)n.

Int n, p;

Float r = 0.12;

Double A;

Cout<<”Enter the value of principle: ”;

Cin>>p;

Cout<<”Enter the number of years: ”;

Cin>>n;

A = p \* pow(1 + r, n);

Cout<<A;

4 \* 6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  | 18 |
|  |  |  |  |  | 24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 30 |
|  |  |  |  |  | 36 |
|  |  |  |  |  | 42 |
|  |  |  |  |  | 48 |
|  | | |  | | |
|  | | |  | | |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

i) Float num[2][4][6]

Size of float: 4 bytes (standard for most systems)

Dimensions: 2 (first dimension) × 4 (second dimension) × 6 (third dimension)

Memory Size=Size of float×DimensionsMemory Size=Size of float×Dimensions

Memory Size=4 bytes×2×4×6Memory Size=4bytes×2×4×6

Memory Size=192 bytesMemory Size=192bytes

Therefore, the estimated memory size for the array num[2][4][6]num[2][4][6] after compilation is 192 bytes.

ii) Char T[4][2][2]

Size of char: 1 byte (standard for most systems)

Dimensions: 4 (first dimension) × 2 (second dimension) × 2 (third dimension)

Memory Size=Size of char×DimensionsMemory Size=Size of char×Dimensions

Memory Size=1 byte×4×2×2Memory Size=1byte×4×2×2

Memory Size=16 bytesMemory Size=16bytes

Therefore, the estimated memory size for the array T[4][2][2]T[4][2][2] after compilation is 16 bytes.

Write a C program to store twenty floating numbers in an array, then display them in reverse order.

Float rev[];

Int I;

For(I = 20; I > 0, i--)

{

Cout<<rev[i];

}