1. Write a program to check if a number is positive, negative, or zero.

|  |  |
| --- | --- |
| **Input** | Reads the integer from the user using scanf |
| **Process** | Describes what the logic is doing (and why) |
| **Output** | Prints the final categorization |

#include <stdio.h>

void main

{

int num=7;

printf("Enter an integer: ");

if (scanf("%d", &num) != 1)

{

printf(stderr, "Invalid input. Please enter a valid integer.\n");

}

if (num > 0)

{

printf("%d is positive.\n", num);

}

else if (num < 0)

{

printf("%d is negative.\n", num);

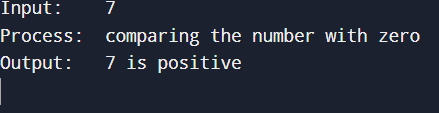
}

else

{

printf("The number is zero.\n");

}

}

1. Write a program to find the largest among three numbers.

|  |  |
| --- | --- |
| **Input** | :Reads **three numbers** (a, b, c) one at a time with prompts. |

|  |  |
| --- | --- |
| **Process** | :Evaluates nested if-else logic: first compares a vs b, then compares the larger one with c to find the maximum. |

|  |  |
| --- | --- |
| **Output** | :Prints the largest number amongst the three, formatted clearly in one line. |

#include <stdio.h>

Void main

{

double a = 13, b = 7, c = 13;

printf("Input‑1: %g\nInput‑2: %g\nInput‑3: %g\n", a, b, c);

printf("Process: comparing the three numbers to find the maximum\n");

if (a >= b)

{

if (a >= c) largest = a;

else largest = c;

}

Else

{

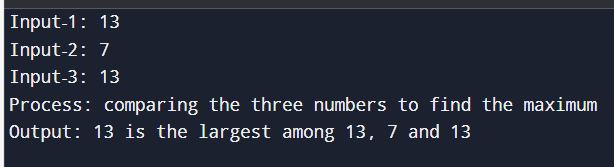
if (b >= c) largest = b;

else largest = c;

}

printf("Output: %.6g is the largest among %.6g, %.6g and %.6g\n", largest, a, b, c);

}



1. Write a program to check if a year is a leap year.

#include <stdio.h>

Void main

{

if (year % 400 == 0)

else if (year % 100 == 0

else if (year % 4 == 0)

}

{

int year = 2024;

printf("Input (year): %d\n", year);

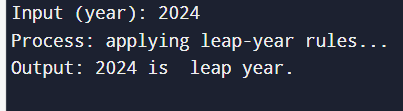
printf("Process: applying leap-year rules...\n");

printf("Output: %d is %s leap year.\n",

year,

checkLeap(year) ? "" : "not a");

}



1. Write a program to check whether a character is a vowel or consonant.

#include <stdio.h>

Void main

{

char ch = 'u';

printf("Input (character): %c\n", ch);

printf("Process: examining whether '%c' is vowel or consonant\n", ch);

if (!isalpha((unsigned char)ch))

{

printf("Output: %c is not an alphabet\n", ch);

}

else if (strchr("aeiouAEIOU", ch))

{

printf("Output: %c is a vowel\n", ch);

}

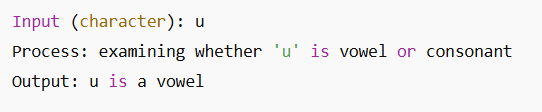
Else

{

printf("Output: %c is a consonant\n", ch);

}

}



1. Write a program to assign grades based on marks.

#include <stdio.h>

Void main

{

int marks = 58;

char grade;

printf("Input (marks out of 100): %d\n", marks);

printf("Process: assigning grade based on marks\n");

if (marks < 0 || marks > 100)

{

printf("Output: Invalid marks (must be 0–100)\n");

}

Else

{

if (marks >= 90) grade = 'A';

else if (marks >= 80) grade = 'B';

else if (marks >= 70) grade = 'C';

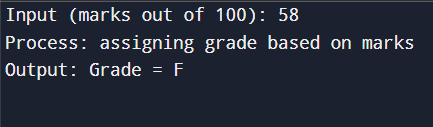
else if (marks >= 60) grade = 'D';

else grade = 'F';

printf("Output: Grade = %c\n", grade);

}

}



1. Write a program to check whether a number is divisible by 5 and 11.

|  |  |
| --- | --- |
| **Input:** | An integer num provided by the user via scanf. |
| **Process** | :1. **Input Validation**: Ensure the input is a valid integer.  2. **Divisibility Check**: Use the modulo operator (%) to check if num is divisible by both 5 and 11.  3. **Logical AND**: Combine the two conditions using the logical AND operator (&&). |
| **Output** | - If num is divisible by both 5 and 11: printf("Number is divisible by 5 and 11");  - Otherwise: printf("Number is not divisible by 5 and 11"); |

#include <stdio.h>

void main

{

int n = 55

printf("Input (integer): %d\n", n);

printf("Process: testing if %d %% 5 == 0 AND %d %% 11 == 0\n", n, n);

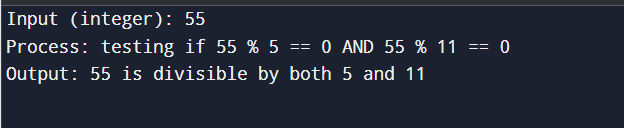
if (n % 5 == 0 && n % 11 == 0)

printf("Output: %d is divisible by both 5 and 11\n", n);

else

printf("Output: %d is NOT divisible by both 5 and 11\n", n);

}



1. Write a program to find the absolute value of a number.

#include <stdio.h>

void main

{

int x = -35

printf("Given value: %d\n", x);

printf("Absolute value: %d\n", abs(x));

}



1. Write a menu-driven program to perform +, -, \*, / operations.

#include <stdio.h>

void main()

{

int choice;

float num1, num2, result;

while (1)

{

printf("\n===== Simple Calculator =====\n");

printf("1. Addition (+)\n");

printf("2. Subtraction (-)\n");

printf("3. Multiplication (\*)\n");

printf("4. Division (/)\n");

printf("5. Exit\n");

printf("Choose an operation (1-5): ");

scanf("%d", &choice);

if (choice == 5)

{

printf("Exiting the program. Goodbye!\n");

break;

}

printf("Enter two numbers: ");

scanf("%f %f", &num1, &num2);

switch (choice)

{

case 1:

result = num1 + num2;

printf("Result: %.2f + %.2f = %.2f\n", num1, num2, result);

break;

case 2:

result = num1 - num2;

printf("Result: %.2f - %.2f = %.2f\n", num1, num2, result);

break;

case 3:

result = num1 \* num2;

printf("Result: %.2f \* %.2f = %.2f\n", num1, num2, result);

break;

case 4:

if (num2 != 0)

{

result = num1 / num2;

printf("Result: %.2f / %.2f = %.2f\n", num1, num2, result);

}

else

{

printf("Error: Division by zero is not allowed.\n");

}

break;

printf("Invalid choice. Please select a valid option (1-5).\n");

}

}

}

1. Write a program to find roots of a quadratic equation.

#include <stdio.h>

void main()

{

float a, b, c;

float discriminant, root1, root2, realPart, imagPart;

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

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

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

if (a == 0)

{

printf("This is not a quadratic equation (a cannot be 0).\n");

}

else if (discriminant > 0)

{

root1 = (-b + sqrt(discriminant)) / (2\*a);

root2 = (-b - sqrt(discriminant)) / (2\*a);

printf("Roots are real and distinct:\n");

printf("Root 1 = %.2f\n", root1);

printf("Root 2 = %.2f\n", root2);

}

else if (discriminant == 0)

{

root1 = -b / (2\*a);

printf("Roots are real and equal:\n");

printf("Root 1 = Root 2 = %.2f\n", root1);

}

Else

{

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

imagPart = sqrt(-discriminant) / (2\*a);

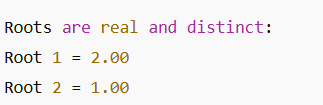
printf("Roots are complex and imaginary:\n");

printf("Root 1 = %.2f + %.2fi\n", realPart, imagPart);

printf("Root 2 = %.2f - %.2fi\n", realPart, imagPart);

}

}



1. Write a program to find the number of digits in a number.

|  |  |
| --- | --- |
| **Input** | Data received from the user or another source |
| **Process** | Logical steps or computations performed on the input |
| **Output** | Result or information displayed to the user |

#include <stdio.h>

void main()

{

int number, count = 0;

printf("Enter an integer: ");

scanf("%d", &number);

if (number == 0)

{

count = 1;

}

Else

{

if (number < 0)

{

number = -number;

}

while (number != 0)

{

number = number / 10;

count++;

}

}

printf("Number of digits: %d\n", count);

}

