

✓ Week-03-Decision Making and Branching - if, if...else and nested if...else, if...else if and switch...case

 Week-03-01-Practice Session-Coding

✓ Done

 Week-03-02-Practice Session-Coding

✓ Done

 Week-03-03-Practice Session-Coding

✓ Done

GE23131-Programming Using C-2024

C++24 Navigation
 [Previous] [Next] [Home]
 Show the page 2 of 4 items
 [Back to top]

Status: Finished
Started: Monday, 23 December 2024, 5:03 PM
Completed: Tuesday, 13 November 2024, 11:18 AM
Duration: 40 days 8 hours

Question 1
 Correct
 Marked out of 100
 100% question

Write a program that determines the name of a shape from its number of sides. List the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should report shapes with anywhere from 3 to 10 sides including 10 sides. The number of sides outside of this range is entered than your program should display an appropriate error message.

Sample Input 1

3

Sample Output 1

Triangle

Sample Input 2

7

Sample Output 2

Heptagon

Sample Input 3

11

Sample Output 3

The number of sides is not supported.

Answer: (provide response 0%)

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n;
6     string s;
7     while (n < 3 || n > 10)
8     {
9         cout << "Enter number of sides: ";
10        cin >> n;
11    }
12    if (n == 3) s = "Triangle";
13    else if (n == 4) s = "Square";
14    else if (n == 5) s = "Pentagon";
15    else if (n == 6) s = "Hexagon";
16    else if (n == 7) s = "Heptagon";
17    else if (n == 8) s = "Octagon";
18    else if (n == 9) s = "Nonagon";
19    else if (n == 10) s = "Decagon";
20    else s = "The number of sides is not supported.";
21    cout << s << endl;
22    return 0;
  
```

Input	Expected	Got
3	Triangle	✓
7	Heptagon	✓
11	The number of sides is not supported.	✓

Reset all input: ✓

Question 2
 Correct
 Marked out of 100
 100% question

The Chinese zodiac assigns animals to years in a 12 year cycle. Over 15 year cycle, a Chinese zodiac is the same. The pattern repeats from 1900 being another year of the Dragon, and 1988 being another year of the Rat.

Year Animal

2008 Dragon

2003 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2006 Rat

2008 Ox

2010 Tiger

2011 Horse

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to years, and just the year listed in the table.

Sample Input 1

2004

Sample Output 1

Monkey

Sample Input 2

2010

Sample Output 2

Tiger

Answer: (provide response 0%)

```

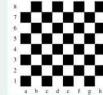
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int year;
6     string animal;
7     while (year < 1900 || year > 2100)
8     {
9         cout << "Enter year: ";
10        cin >> year;
11    }
12    if (year % 12 == 0) animal = "Dragon";
13    else if (year % 12 == 1) animal = "Snake";
14    else if (year % 12 == 2) animal = "Horse";
15    else if (year % 12 == 3) animal = "Sheep";
16    else if (year % 12 == 4) animal = "Monkey";
17    else if (year % 12 == 5) animal = "Rooster";
18    else if (year % 12 == 6) animal = "Dog";
19    else if (year % 12 == 7) animal = "Pig";
20    else if (year % 12 == 8) animal = "Rat";
21    else if (year % 12 == 9) animal = "Ox";
22    else if (year % 12 == 10) animal = "Tiger";
23    else if (year % 12 == 11) animal = "Horse";
24    cout << animal << endl;
25    return 0;
  
```

Input	Expected	Got
2004	Monkey	✓
2010	Tiger	✓

Reset all input: ✓

Question 3
 Correct
 Marked out of 100
 100% question

Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below.



Write a program that reads a position from the user. Use an if statement to determine if the column letter is a black square or a white square. Then convert the number to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d7 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

a1

Sample Output 1

The square is black.

Sample Input 2

d5

Sample Output 2

The square is white.

Answer: (provide response 0%)

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     char c;
6     int r;
7     if (c < 'a' || c > 'h' || r < 1 || r > 8)
8     {
9         cout << "Invalid position. Please enter a valid position." << endl;
10        return 0;
11    }
12    if ((c - 'a' + r) % 2 == 0)
13        cout << "The square is black." << endl;
14    else
15        cout << "The square is white." << endl;
16    return 0;
  
```

Input	Expected	Got
a1	The square is black.	✓
d5	The square is white.	✓

Reset all input: ✓