

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

Week-03-01-Practive Session-coding

ROLL NO: 241801294

NAME: THARUN N

Q1)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Wednesday, 6 November 2024, 2:32 PM
Duration	47 days 3 hours

Question **1**

Correct

Marked out of 3.00

Flag question

Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false.
Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 1 25 53 Sample Output 1 false Sample Input 2 27 77 Sample Output 2 true

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int num1, num2;
5     scanf("%d %d", &num1,&num2);
6     int lastDigit1 = num1 % 10;
7     int lastDigit2 = num2 % 10;
8     if(lastDigit1 == lastDigit2) {
9         printf("true\n");
10    }
11    else{
12        printf("false\n");
13    }
14 }
```

	Input	Expected	Got	
✓	25 53	false	false	✓
✓	27 77	true	true	✓


Passed all tests! ✓

Q2)

Question **2**

Correct

Marked out of
5.00

 Flag question

Objective

In this challenge, we're getting started with conditional statements.

Task

Given an integer, n , perform the following conditional actions:

- If n is odd, print **Weird**
- If n is even and in the inclusive range of **2** to **5**, print **Not Weird**
- If n is even and in the inclusive range of **6** to **20**, print **Weird**
- If n is even and greater than **20**, print **Not Weird**

Complete the stub code provided in your editor to print whether or not n is weird.

Input Format

A single line containing a positive integer, n .

Constraints

- $1 \leq n \leq 100$

Output Format

Print **Weird** if the number is weird; otherwise, print **Not Weird**.

Sample Input 0

3

Sample Output 0

Weird

Sample Input 1

24

Sample Output 1

Not Weird

Explanation

Sample Case 0: $n = 3$

n is odd and odd numbers are weird, so we print **Weird**.

Sample Case 1: $n = 24$

$n > 20$ and n is even, so it isn't weird. Thus, we print **Not Weird**.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d", &n);
6     if (n % 2 != 0)
7     {
8         printf("Weird\n");
9     }
10    else
11    {
12        if (n >= 2 && n <= 5)
13        {
14            printf("Not Weird\n");
15        }
16        else if (n >= 6 && n <= 20)
17        {
18            printf("Weird\n");
19        }
20        else if (n > 20)
21        {
22            printf("Not Weird");
23        }
24    }
25 }
26 }
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓
✓	24	Not Weird	Not Weird	✓

Passed all tests! ✓

Q3)

Question **3**

Correct

Marked out of
7.00

Flag question

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3, 5 and 4 form a Pythagorean triple, since $3^2 + 4^2 = 25 = 5^2$. You are given three integers, a , b , and c . They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters. Sample Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a, b, c;
5     scanf("%d %d %d", &a,&b,&c);
6     if((a*a+b*b==c*c) ||
7        (a*a+c*c==b*b) ||
8        (b*b+c*c==a*a)){
9         printf("yes\n");
10    }
11    else{
12        printf("no\n");
13    }
14 }
```

	Input	Expected	Got	
✓	3 5 4	yes	yes	✓
✓	5 8 2	no	no	✓

Passed all tests! ✓

Week-03-02-Practive Session-coding

ROLL NO: 241801294

NAME: THARUN N

Q1)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 9 November 2024, 9:52 AM
Duration	44 days 7 hours

Question **1**
Correct
Marked out of 3.00
[Flag question](#)

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then 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: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int sides;
5     scanf("%d", &sides);
6     switch (sides)
7     {
8         case 3:
9             printf("Triangle");
10            break;
11            case 4:
12                printf("Quadrilatula");
13                break;
14                case 5:
15                    printf("Pentagon");
16                    break;
17                    case 6:
18                        printf("Hexagon");
19                        break;
20                        case 7:
21                            printf("Heptagon");
22                            break;
23                            case 8:
24                                printf("Octagon");
25                                break;
26                                case 9:
27                                    printf("Nonagon");
28                                    break;
29                                    case 10:
30                                        printf("Decagon");
31                                        break;
32                                    default:
33                                        printf("The number of sides is not supported.\n");
34                                        break;
35                                }
36    }
```

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

Passed all tests! ✓

Q2)

Question 2

Correct

Marked out of 5.00

Flag question

The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the Dragon, and 1999 being another year of the Hare.

Year	Animal
2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

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 zero, not just the ones listed in the table.

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 zero, not just the ones listed in the table.

Sample Input 1

2004

Sample Output 1

Monkey

Sample Input 2

2010

Sample Output 2

Tiger

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int year;
5     char * animals[] = {
6         "Dragon", "Snake", "Horse", "Sheep", "Monkey", "Rooster", "Dog", "Pig", "Rat", "Ox", "Tiger", "Horse"
7     };
8     scanf("%d", &year);
9     int index = (year - 2000) % 12;
10    if (index < 0 ) {
11        index += 12;
12    }
13    printf("%s\n", animals[index]);
14    return 0;
15 }
```

	Input	Expected	Got	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

Passed all tests! ✓

Q3

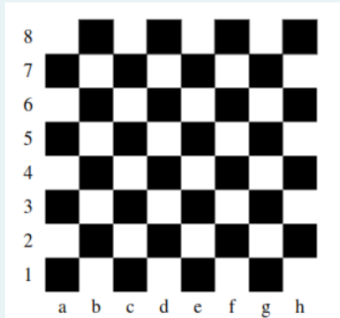
Question 3

Incorrect

Marked out of 7.00

Flag 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 begins with a black square or a white square. Then use modular arithmetic 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 d5 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

a 1

Sample Output 1

The square is black.

Sample Output 2

The square is white.

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     char a;
5     int b;
6     scanf("%c", &a);
7     scanf("%d", &b);
8
9     switch (a) {
10        case 'a':
11        case 'c':
12        case 'e':
13        case 'g':
14            if (b%2 == 0) {
15                printf("The square is white.");
16            }else{
17                printf("The square is black.");
18            }
19            break;
20        default:
21            if (b%2 == 0) {
22                printf("The square is black.");
23            }else{
24                printf("The square is white.");
25            }
26            break;
27    }
28 }
```

	Input	Expected	Got	
✓	a 1	The square is black.	The square is black.	✓
✓	d 5	The square is white.	The square is white.	✓

Passed all tests! ✓

Array Applications

ROLL NO: 241801294

NAME: THARUN N

Q1)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 16 November 2024, 11:39 PM
Duration	36 days 17 hours

Question 1

Correct

Marked out of 1.00

Flag question

Write a program to find the largest and second largest elements with in the elements of the given one dimensional array.

For example, if the user gives the **input** as:

5

Next, the program should print the messages one by one on the console.

if the user gives the **input** as:

10
50
30
20
25

then the program should **print** the result as:

The largest element of the array = 50
The second largest element of the array = 30

Hints

Let us assume first element it self as the **large**, **second_large** and then compare both with all the other elements.
If any one found as **large** then change the value of the large, otherwise compare it with **second_large** and exchange it when necessary condition is satisfied.

For example:

Input	Result
5 2 5 9 7 3	The largest element of the array = 9 The second largest element of the array = 7
6 64 87 34 58 62 18	The largest element of the array = 87 The second largest element of the array = 64

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<limits.h>
3 int main(void)
4 {
5     int v;
6     scanf("%d",&v);
7     int arr[v];
8     if(v<2)
9     {
10         printf("Number of elements should be greater than zero");
11         return 0;
12     }
13     for(int i=0;i<v;i++)
14     {
15         scanf("%d",&arr[i]);
16     }
17     int lNum=INT_MIN,s1Num=INT_MIN;
18     for(int i=0;i<v;i++)
19     {
20         if(lNum<arr[i])
21         {
22             s1Num=lNum;
23             lNum=arr[i];
24         }
25         else if(s1Num<arr[i])
26         {
27             s1Num=arr[i];
28         }
29     }
30     printf("The largest element of the array = %d\nThe second largest element of the array = %d",lNum,
```

	Input	Expected	Got	
✓	5 2 5 9 7 3	The largest element of the array = 9 The second largest element of the array = 7	The largest element of the array = 9 The second largest element of the array = 7	✓
✓	6 64 87 34 58 62 18	The largest element of the array = 87 The second largest element of the array = 64	The largest element of the array = 87 The second largest element of the array = 64	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

Flag question

Write a program to find the minimum and second minimum elements with in the elements of one dimensional array.

Constraints:

- $1 \leq N \leq 10^3$
- $1 \leq \text{Elements of the array} \leq 10^6$

Instruction: To run your custom test cases strictly map your input and output layout with the visible test cases.

Hints

Let us assume that first element it self as the **minimum**, **second_minimum** and then compare both with all the other elements. If any one found as **minimum** then change the value of the minimum, otherwise compare it with **second_minimum** and exchange it when necessary condition is satisfied.

For example:

Input	Result
4	Min element = 32
65 32 85 96	Second min element = 65

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<limits.h>
3 int main() {
4     int v;
5     scanf("%d", &v);
6     int arr[v];
7     for (int i = 0; i < v; i++) {
8         scanf("%d", &arr[i]);
9     }
10    int n1 = INT_MAX, n2 = INT_MAX;
11    for (int i = 0; i < v; i++) {
12        if (arr[i] < n1) {
13            n2 = n1;
14            n1 = arr[i];
15        } else if (arr[i] < n2) {
16            n2 = arr[i];
17        }
18    }
19    printf("Min element = %d\nSecond min element = %d", n1, n2);
20 }
```

	Input	Expected	Got	
✓	4 65 32 85 96	Min element = 32 Second min element = 65	Min element = 32 Second min element = 65	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Flag question

Write a program to read a student n subjects marks in an array and find the total, average of the marks.

For example, if the user gives the **input** as:
3

Next, the program should print the messages one by one on the console.
if the user gives the **input** as:
75
80
85

then the program should **print** the result as:

The total marks = 240
The average marks = 80.000000

Hints
marks are integers, **total** is also an integer but **average** is a float value, so typecast it.

For example:

Input	Result
5 45 65 55 75 85	The total marks = 325 The average marks = 65.000000
4 36 45 38 56	The total marks = 175 The average marks = 43.750000

```
1 #include<stdio.h>
2 int main () {
3     int v;
4     scanf("%d", &v);
5     int t = 0;
6     for (int i = 0; i<v; i++) {
7         int x;
8         scanf("%d", &x);
9         t+=x;
10    }
11    float x = (t*1.0)/v;
12    printf("The total marks = %d\nThe average marks = %f", t, x);
13 }
```

	Input	Expected	Got	
✓	5 45 65 55 75 85	The total marks = 325 The average marks = 65.000000	The total marks = 325 The average marks = 65.000000	✓
✓	4 36 45 38 56	The total marks = 175 The average marks = 43.750000	The total marks = 175 The average marks = 43.750000	✓

Problem solving with Strings

ROLL NO: 241801294

NAME: THARUN N

Q1)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 19 November 2024, 10:21 PM
Duration	33 days 19 hours

Question 1

Correct

Marked out of 1.00

Flag question

Fill in the missing code in the below sample code which counts the number of vowels, consonants, digits and spaces are presented in a given string.

Initially, the variables vowels, consonants, digits and spaces are initialized to 0.

Iterate the string from the **first** character to **last** character to find all vowels, consonants, digits and spaces.

When a vowel character is found, vowel variable is incremented by 1. Similarly, consonants, digits and spaces are incremented when these characters are found in the string.

Finally, the count is displayed on the screen.

For example:

Input	Result
kohli hits 100 in every cricket match!	Vowels = 9 Consonants = 19 Digits = 3 White spaces = 6

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2 #include<ctype.h>
3 int main()
4 {
5     char line[100];
6     int i, vowels = 0, consonants = 0, digits = 0, spaces = 0;
7     fgets(line, sizeof(line), stdin);
8     for (i = 0; line[i] != '\0'; i++)
9     { // Complete the code in for
10         if (line[i] == 'a' || line[i] == 'A' || line[i] == 'e' || line[i] == 'E' || line[i] == 'i' ||
11         { // Write the condition part
12             ++vowels;
13         }
14         else if (isalpha(line[i]))
15         { // Write the condition part
16             ++consonants;
17         }
18         else if (isdigit(line[i]))
19         { // Write the condition part
20             ++digits;
21         }
22         else if (line[i] == ' ')
23         { // Write the condition part
24             ++spaces;
25         }
26     }
27     printf("Vowels = %d\n", vowels);
28     printf("Consonants = %d\n",consonants);
29     printf("Digits = %d\n",digits);
30     printf("White spaces = %d", spaces);
31     return 0;
32 }
```

	Input	Expected	Got	
✓	kohli hits 100 in every cricket match!	Vowels = 9 Consonants = 19 Digits = 3 White spaces = 6	Vowels = 9 Consonants = 19 Digits = 3 White spaces = 6	✓

Passed all tests! ✓

Q2)

Question **2**

Correct

Marked out of 1.00

Flag question

Fill in the missing code in the below sample code which copies a given string into another string.

Initially, read a string from the standard input device and write a loop to copy each character of given string into another string till the end of the string is reached.

Place '\0' at the end of the copied string.

Finally, the copied string is displayed on the screen.

For example:

Input	Result
GangaRiver	The copied string = GangaRiver

Answer: (penalty regime: 0 %)

Reset answer

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char str1[50], str2[50];
6      int i;
7      scanf("%s", str1);
8      for (i = 0; str1[i] != '\0' ; i++ )
9      { //Complete the code in for
10         str2[i] = str1[i];
11     }
12     str2[i] = '\0'; //Complete the statement
13     printf("The copied string = %s\n", str2);
14     return 0;
15 }
```

	Input	Expected	Got	
✓	GangaRiver	The copied string = GangaRiver	The copied string = GangaRiver	✓

Passed all tests! ✓

Q3)

Question **3**

Correct

Marked out of 1.00

Flag question

Fill in the missing code in the below sample code which concatenates two given strings and store the result in another string.

Read two strings from the standard input device and write a loop to copy each character of the first string into third string till the end of the first string.

Write another loop to copy each character of the second string into third string till the end of second string.

Now place '\0' at the end of the third string.

Finally, display the third string.

For example:

Input	Result
Narendra Modi	NarendraModi

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char a[20], b[20], c[20];
6     int i, j;
7     scanf("%s", a);
8     scanf("%s", b);
9     for (i = 0; a[i] != '\0'; i++)
10 { // Complete the code in for
11     c[i] = a[i]; //Complete the statement
12 }
13     for (j = 0; b[j] != '\0'; j++ )
14 { // Complete the code in for
15     c[i] = b[j]; //Complete the statement
16     i++;
17 }
18     c[i] = '\0'; //Complete the statement
19     printf("%s\n", c);
20     return 0;
21 }
```

	Input	Expected	Got	
✓	Narendra Modi	NarendraModi	NarendraModi	✓

Passed all tests! ✓

Q4)

Question **4**

Correct

Marked out of
1.00

Flag question

Fill in the missing code in the below sample code to check whether the given two strings are equal or not.

Read two strings from the standard input device and write a loop to check each character of the first string with second string till the end of the first string is reached.

If any character is not equal then break the loop and say "**Two strings are not equal**".

If all the characters are equal and the length of two strings is also equal then display "**Two strings are equal**".

For example:

Input	Result
Godavari Godavari	Two strings are equal
Narmada narmada	Two strings are not equal

```

1 #include <stdio.h>
2
3 int main()
4 {
5     char a[20], b[20];
6     int i = 0, flag = 1;
7     scanf("%s", a);
8     scanf("%s", b);
9     while (a[i] != '\0')
10 { //Complete the condition part
11     if (a[i] != b[i])
12     { //Complete the condition part
13         flag = 0; //Complete the statement
14         break;
15     }
16     i++;
17 }
18 if (flag == 1)
19 { //Complete the condition part
20     printf("Two strings are equal\n");
21 }
22 else
23 {
24     printf("Two strings are not equal\n");
25 }
26 return 0;
27 }

```

	Input	Expected	Got	
✓	Godavari Godavari	Two strings are equal	Two strings are equal	✓
✓	Narmada narmada	Two strings are not equal	Two strings are not equal	✓

Passed all tests! ✓

Q5)

Question 5

Correct

Marked out of
1.00

Flag question

Fill in the missing code in the below sample code to search the occurrence of a given character in a given string.

Read a string and a character from the standard input device and write a loop to check each character of the string with a given character.

If the given character is equal to a character in the string then increment the count with in the loop.

Finally, display the count variable which has the total number of occurrences of the given character.

For example:

Input	Result
CurrencyDemonitisation n	Occurence of character 'n' in the given string CurrencyDemonitisation = 3

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <stdio.h>
2
3 int main()
4 {
5     char str[20], ch;
6     int count = 0, i;
7     scanf("%s", str);
8     scanf(" %c", &ch);
9     for (i = 0; str[i] != '\0'; i++)
10 { // Complete the code in for
11     if (str[i] == ch)
12     { // Write the condition part
13         count++;
14     }
15 }
16 if (count == 0)
17 { // Write the condition part
18     printf("The character '%c' is not presented in the string %s\n", ch, str);
19 }
20 else
21 {
22     printf("Occurence of character '%c' in the given string %s = %d\n", ch, str, count);
23 }
24 return 0;
25 }

```

	Input	Expected	Got
✓	CurrencyDemonitisation n	Occurence of character 'n' in the given string CurrencyDemonitisation = 3	Occurence of

Passed all tests! ✓

Q6)

Question 6
Correct
Marked out of 1.00
Flag question

Fill in the missing code in the below sample code to count total number of uppercase and lowercase characters from the accepted string.

Read a string from the standard input device and write a loop to check each character, whether it is uppercase or lowercase of the given string.

If the given character is uppercase then increment the upper_count with in the loop.

If the given character is lowercase then increment the lower_count with in the loop.

Finally display the upper_count and lower_count.

For example:

Input	Result
KrishnaAndGodavariAneRivers	Number of uppercase letters = 5 Number of lowercase Letters = 22

Answer: (penalty regime: 0 %)

Reset answer


```

1 #include<stdio.h>
2
3 int main()
4 {
5     int upper_count = 0, lower_count = 0;
6     char ch[80];
7     int i;
8     scanf("%s", ch ); // Complete the statement
9     i = 0; // Complete the statement
10    while (ch[i] != '\0' )
11    { // Write the condition part
12        if (ch[i] >= 'A' && ch[i] <= 'Z')
13        { // Write the condition part
14            upper_count++;
15        }
16        if (ch[i] >= 'a' && ch[i] <= 'z')
17        { // Write the condition part
18            lower_count++;
19        }
20        i++;
21    }
22    printf("Number of uppercase letters = %d\n", upper_count);
23    printf("Number of lowercase Letters = %d\n", lower_count);
24    return 0;
25 }

```

	Input	Expected	Got	
✓	KrishnaAndGodavariAreRivers	Number of upprcase letters = 5 Number of lowercase Letters = 22	Number of uppercase letters = 5 Number of lowercase Letters = 22	✓

Passed all tests! ✓

Q7)

Question 7
Correct

Marked out of
1.00

Flag question

Fill in the missing code in the below sample code to reverse the given string.

Hints

Step:1 Read a string from the standard input device.

Step:2 Write a loop to find the length of the string.

Step:3 Write another loop to interchange the characters from first to last of the string.

Step:4 Finally display the reverse of a string.

For example:

Input	Result
Software	The reverse of a given string : erawtfos

```

1 #include<stdio.h>
2
3 int main()
4 {
5     char ch[80], temp[80];
6     int i, j;
7     scanf("%s", ch);
8     i = j = 0;
9     while (ch[j] != '\0')
10    { // Write the condition part
11        j++;
12    }
13    j--;
14    while (j >= 0 )
15    { // Write the condition part
16        temp[i] = ch[j]; // Complete the statement
17        i++;
18        j--;
19    }
20    printf("The reverse of a given string : %s\n", temp);
21    return 0;
22 }

```

	Input	Expected	Got	
✓	Software	The reverse of a given string : erawtfos	The reverse of a given string : erawtfos	✓

Passed all tests! ✓

Q8)

Question 8

Correct

Marked out of 1.00

Flag question

Fill in the missing code in the below sample code to check whether the given string is a palindrome or not.

Read a string from the standard input device and write a loop to check the characters of the given string with the reverse string.

If all the characters are equal then display **"The given string is a palindrome"**, otherwise display **"The given string is not a palindrome"**.

For example:

Input	Result
12321	The given string 12321 is a palindrome
amaravathi	The given string amaravathi is not a palindrome

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char ch[80];
6     int i, j, length, flag = 0;
7     scanf("%s", ch); // Complete the statement
8     length = 0;
9     while (ch[length] != '\0')
10 { //Write the condition part
11     length++;
12 }
13 j = length-1;
14 for (i =0; i < length; i++)
15 { // Complete the code in for
16     if (ch[i] != ch[j])
17     { // Write the condition part
18         flag++;
19         break;
20     }
21     j--;
22 }
23 if (flag == 0)
24 { // Write the condition part
25     printf("The given string %s is a palindrome\n", ch); // Complete the statement
26 }
27 else
28 {
29     printf("The given string %s is not a palindrome\n", ch ); // Complete the statement
30 }
31 return 0;
32 }
```

	Input	Expected	Got
✓	12321	The given string 12321 is a palindrome	The given string 12321 is a palindrome ✓
✓	amaravathi	The given string amaravathi is not a palindrome	The given string amaravathi is not a palindrome ✓

Passed all tests! ✓

String manipulation functions

ROLL NO: 241801294

NAME: THARUN N

Q1)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 19 November 2024, 10:32 PM
Duration	33 days 19 hours

Question 1

Correct

Marked out of 1.00

Flag question

In C language, we have four types of string functions that are used for performing **string operations**. They are strlen(), strcpy(), strcat(), strcmp().

The function strlen() is used to find the **length** of the given string. This function returns only the **integer data** (or) **numeric data**.

The function strlen() counts the number of characters in a given string and returns the integer value.

It stops counting the character when **NULL** character is found. Because, **NULL** character indicates the end of the string in C.

The syntax of strlen() is integer_variable = strlen(string);.

Here string is a group of characters, strlen() function finds the **length** of the string and the **integer** value will be stored in the integer_variable.

The string.h header file supports all the string functions in C language.

Fill in the missing code in the below program to find the **length** of a string using **strlen()** function.

For example:

Input	Result
NarendraModi	The length of the string NarendraModi is 12

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char ch[20];
7     scanf("%s", ch);
8     printf("The length of the string %s is %ld\n", ch, strlen(ch)); //Correct the code
9     return 0;
10 }
```

	Input	Expected	Got	
✓	NarendraModi	The length of the string NarendraModi is 12	The length of the string NarendraModi is 12	✓

Passed all tests! ✓

Q2)

Question 2

Correct

Marked out of 1.00

Flag question

The function strcpy() is used to **copy** one string into another string including the NULL character (terminator char '\0').

The syntax of strcpy() is strcpy(string1, string2);.

Where string1, string2 are two strings and the string2 is copied into string1. In this case the copied string is available in string1 and both strings contains the same data.

If the length of string1 is less than the length of string2 then entire string2 value will not be copied into string1.

For example, consider the length of string1 is **20** and the length of string2 is **30**. Then, only the first **20** characters from string2 will be copied into string1, the remaining **10** characters will not be copied and will be **truncated**.

Understand and retype the below code which demonstrates the usage of **strcpy()** function.

```
#include <stdio.h>
#include <string.h>

int main()
{
    char str1[20], str2[20];
    scanf("%s", str2);
    strcpy(str1, str2);
    printf("The copied string = %s", str1);
    return 0;
}
```

For example:

Input	Result
Rose	The copied string = Rose

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char a[20], b[20];
6     scanf("%s", b);
7     strcpy(a, b);
8     printf("The copied string = %s", a);
9 }
```

	Input	Expected	Got	
✓	Rose	The copied string = Rose	The copied string = Rose	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Flag question

The function `strcat()` is used to concatenate two strings into a single string.

The syntax of `strcat()` is `strcat(string1, string2);`.

where `string1`, `string2` are two different strings. Here `string2` is concatenated with `string1`, and the **concatenated string** is stored in `string1`.

In `strcat()` operation, **NULL character** (`'\0'`) of `string1` is **overwritten** by first character of `string2` and **NULL character** (`'\0'`) is appended (added) at the end of **new** `string1` which is created after `strcat()` operation.

Fill the missing code in the below program to display the **concatenated** string using **`strcat()`** function.

For example:

Input	Result
REC Chennai	RECChennai

Reset answer

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char str1[20], str2[20];
7     scanf("%s", str1);
8     scanf("%s", str2);
9     strcat(str1, str2);
10    //Concat str2 with str1
11    printf("%s\n", str1); // Correct the code
12    return 0;
13 }
```

	Input	Expected	Got	
✓	REC Chennai	RECChennai	RECChennai	✓

Passed all tests! ✓

Q4)

Question 4

Correct

Marked out of 1.00

Flag question

The function strcmp() is used for comparison of two strings and it always returns the numeric data. This function compares strings character by character using their ASCII values.

The syntax of strcmp() is variable_name = strcmp (string1, string2);.

Where string1, string2 are two strings and the variable is of **integer** datatype.

The comparison of two strings is dependent on the **alphabets (characters)** and not on the size (length) of the strings.

If the function strcmp() returns zero, both strings are **equal**.
If the function strcmp() returns a value which is less than zero, **string2** is higher than **string1** (because the **ASCII value** of first unmatched character of **string1** is less than the **ASCII value** of the corresponding character in **string2**)
If the function strcmp() returns a value which is greater than zero, **string1** is higher than **string2** (because the **ASCII value** of first unmatched character of **string1** is greater than the **ASCII value** of the corresponding character in **string2**)

Fill the missing code in the below program to compare two strings using **strcmp()** function.

For example:

Input	Result
NarendraModi narendramodi	The string narendramodi is higher than the string NarendraModi
Krishna Godavari	The string Krishna is higher than the string Godavari
REC REC	The given two strings are equal

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char a[20], b[20];
7
8     scanf("%s", a);
9     scanf("%s", b);
10    //Compare two strings
11
12    if (strcmp(a, b) == 0)
13    { // Correct the code
14        printf("The given two strings are equal\n");
15    }
16    else if (strcmp(a, b) > 0)
17    { // Correct the code
18        printf("The string %s is higher than the string %s\n", a, b);
19    }
20    else
21    {
22        printf("The string %s is higher than the string %s\n", b, a);
23    }
24    return 0;
25 }
```

	Input	Expected	Got
✓	NarendraModi narendramodi	The string narendramodi is higher than the string NarendraModi	The string narendramodi is high
✓	Krishna Godavari	The string Krishna is higher than the string Godavari	The string Krishna is higher th
✓	REC REC	The given two strings are equal	The given two strings are equal

Passed all tests! ✓

