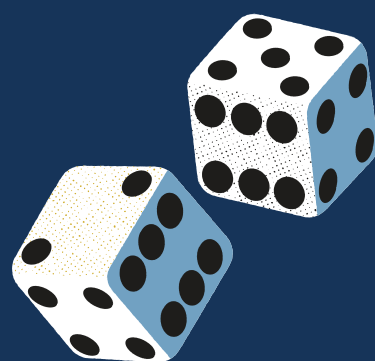


C PROGRAMMING PROJECT



main.c

DICE ROLLING PROGRAM...



By Mahek Trivedi

INDEX

1. NEED FOR PROJECT
2. USER MANUAL
3. MENU TREE
4. SOURCE CODE
5. SOURCE CODE EXPLANATION
6. OUTPUT
7. BIBLIOGRAPHY
8. REFLECTION

NEED FOR PROJECT

1. HELPS TO CREATE AN UNDERSTANDING OF THE FUNDAMENTALS OF PROBABILITY AND RANDOMNESS.
2. HELPS TO DEVELOP A FOUNDATION IN ANALYZING REAL-WORLD SITUATIONS SINCE CONCEPTS OF PROBABILITY AND RANDOMNESS ARE CRUCIAL IN VARIOUS FIELDS.
3. HELPS BEGINNERS IN C PROGRAMMING LANGUAGE TO ENHANCE THEIR BASIC KNOWLEDGE AND SKILLS.
- DEVELOPS RESEARCH AND TRIAL AND ERROR(EXPERIMENTATION) SKILLS.

USER'S MANUAL

HARDWARE USED :

- MODERN OPERATING SYSTEM:
- WINDOWS 7 OR 10
- MAC OS X 10.11 OR HIGHER, 64-BIT
- LINUX: RHEL 6/7, 64-BIT (ALMOST ALL LIBRARIES ALSO WORK IN UBUNTU)
- *86 64-BIT CPU (INTEL / AMD ARCHITECTURE)
- 4 GB RAM
- 5 GB FREE DISK SPACE

SOFTWARE REQUIRED:

- C PROGRAMMING (REPLIT)

MENU TREE

DICE ROLLING PROGRAM

FACES

Faces < 25 && Faces > 1

THORWS

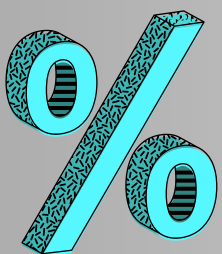
Throws < 500 && Throws > 1

{SRAND}

srand(time(0));
To Set seed based on
current time

Calculate percentage without counting same number more than once for (loop = 0; loop < Faces; loop++) {count = 0;

```
for (j = 0; j < Throws; j++) {  
    if (arr[j] == loop + 1) {  
        count++; arr[j] = 0; } }  
a = (count * 100) / Throws;
```



SOURCE CODE

```
main.c x +
main.c > f main

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(void) {
6      // Declaring the variables
7      int Faces, Throws, i, num, j, count, a;
8      char input[100]; // Buffer for handling non-integer input
9      int *arr;
10
11     // Loop to enter faces and check if the input is valid
12     while (1) {
13         printf("Enter the number of faces:");
14         if (scanf("%d", &Faces) != 1) {
15             // If input is not an integer, we clear the input buffer
16             scanf("%s", input);
17             printf("Invalid Input. Enter again! \n");
18         } else if (Faces < 25 && Faces > 1) {
19             break;
20         } else {
21             printf("Invalid Input. Enter again! \n");
```

```
main.c x +
main.c > f main

22     }
23 }
24
25 // Loop to enter throws and check if the input is valid
26 while (1) {
27     printf("Enter the number of throws:");
28     if (scanf("%d", &Throws) != 1) {
29         // If input is not an integer, we clear the input buffer
30         scanf("%s", input);
31         printf("Invalid Input. Enter again! \n");
32     } else if (Throws < 500 && Throws > 1) {
33         break;
34     } else {
35         printf("Invalid Input. Enter again! \n");
36     }
37 }
38
39 // Allocate memory for the array
40 arr = malloc(Throws * sizeof(int));
41
```



```
main.c × +
main.c > f main

42 // Loop for generating random numbers
43 printf("Generating Throws...\n");
44 srand(time(0)); // Set seed based on current time
45 for (i = 0; i < Throws; i++) {
46     num = rand() % Faces + 1;
47     printf("%d\n", num);
48     arr[i] = num; // Store the value in the array
49 }
50
51 // Calculate percentage without counting same number more than once
52 for (j = 1; j <= Faces; j++) {
53     count = 0;
54     for (i = 0; i < Throws; i++) {
55         if (arr[i] == j) {
56             count++;
57         }
58     }
59     a = (count * 100) / Throws;
60     printf("The percentage of %d = %d%%\n", j, a);
61 }
```

```
main.c × +
main.c > f main

62
63 // Free the memory allocated for the array
64 free(arr);
65
66 return 0;
67 }
```

SOURCE CODE EXPLANATION

1) Libraries used:

- `stdio.h` (Standard Input Output- it provided us with functions for input and output like `printf()` and `scanf()`)
- `stdlib.h` (Standard Library- provides us with the random function. It also has functions related to memory allocation and other general functions.
- `time.h` (Time and Date functions-it has functions related to the time operator. Here, it seeds the random function with the current time (`srand(time(0))`))

2) There is only one function(`main(void)`) used in the code.



SOURCE CODE EXPLANATION

1. The main function declares several variables including Faces, Throws, i, num, j, count, and a. It also declares an array pointer arr and a character array input to handle non-integer input. Here, a pointer to an array is declared to store a sequence of random numbers that will be generated.
2. The program has loops to input the number of faces and the number of throws, ensuring that the input is valid.




SOURCE CODE EXPLANATION

3. The malloc function is used to allocate memory in C. In this case, Throws * sizeof(int) calculates the total number of bytes needed to store the integers based on the number of throws. This dynamically allocated memory allows the program to store the random numbers without having to specify the exact size of the array when the program is compiled.

4. It generates random numbers and stores them in the array, calculates percentages, and prints them. Then it frees the memory allocated for the array.

OUTPUT

```
>_ Console [trash] [x] +  
  [v] Run  
Enter the number of faces:L  
Invalid Input. Enter again!  
Enter the number of faces:123  
Invalid Input. Enter again!  
Enter the number of faces:AB  
Invalid Input. Enter again!  
Enter the number of faces:12345  
Invalid Input. Enter again!  
Enter the number of faces:12  
Enter the number of throws:A  
Invalid Input. Enter again!  
Enter the number of throws:6  
Generating Throws...  
2  
5  
5  
5  
11  
11  
The percentage of 1 = 0%  
The percentage of 2 = 16%  
The percentage of 3 = 0%  
The percentage of 4 = 0%  
The percentage of 5 = 50%  
The percentage of 6 = 0%  
The percentage of 7 = 0%  
The percentage of 8 = 0%  
The percentage of 9 = 0%  
The percentage of 10 = 0%  
The percentage of 11 = 33%  
The percentage of 12 = 0%
```

>_ Console   

✓ Run

```
Enter the number of faces:6
Enter the number of throws:4
Generating Throws...
2
2
4
4
The percentage of 1 = 0%
The percentage of 2 = 50%
The percentage of 3 = 0%
The percentage of 4 = 50%
The percentage of 5 = 0%
The percentage of 6 = 0%
```

✓ Run

```
Enter the number of faces:6
Enter the number of throws:4
Generating Throws...
1
5
3
3
The percentage of 1 = 25%
The percentage of 2 = 0%
The percentage of 3 = 50%
The percentage of 4 = 0%
The percentage of 5 = 25%
The percentage of 6 = 0%
```

BIBLIOGRAPHY

1. STACK OVERFLOW
2. FREE CODE CAMP
3. PROGRAMIZ
4. CST 1500 LAB BOOK

REFLECTION

THIS COURSEWORK HAS BEEN INSTRUMENTAL IN FACILITATING A COMPREHENSIVE REVIEW OF THE FUNDAMENTAL CONCEPTS SURROUNDING PROBABILITY AND RANDOMNESS. THROUGH ITS EXECUTION, WE FOUND OURSELVES DELVING DEEPER INTO THE INTRICATE NUANCES OF THESE PRINCIPLES, ALLOWING US TO GAIN A NUANCED UNDERSTANDING BEYOND THE THEORETICAL REALM. LEVERAGING THE FOUNDATIONAL KNOWLEDGE ACQUIRED FROM OUR C PROGRAMMING LECTURES, WE NOT ONLY APPLIED THE BASICS BUT ALSO DELVED FURTHER INTO ADVANCED TECHNIQUES, BROADENING OUR PROGRAMMING EXPERTISE. THE PROCESS WAS NOT SOLELY CONFINED TO WHAT WAS COVERED IN CLASS; IT EXTENDED TO OUR SELF-DRIVEN RESEARCH ENDEAVORS, ENRICHING OUR UNDERSTANDING AND PROFICIENCY IN HANDLING COMPLEX ALGORITHMS AND LOGICAL STRUCTURES.



THANK
YOU