Universidade do Vale do Itajaí
Escola do Mar Ciência e Tecnologia
NEI - EMCT
Disciplina: Algoritmos e Programação II
Professor: Rodrigo Lyra

Trabalho 1.1

Acadêmico (a):_____ **Data**: 14/03/2022 **Conceito**: _____

Confirmando Probabilidades

Vemos muitas probabilidades no dia a dia, mas elas estão corretas? Com o uso de algoritmos podemos tirar isso a limpo, fazer testes e verificar o que está acontecendo. Lançamos um dado de seis lados 10 vezes e cai 5 vezes o número 5, o dado é viciado? as probabilidades estão erradas? Podemos jogar o dado 10 mil vezes e tirar a prova real, no mundo real isso é improvável, mas em algoritmos...

Podemos gerar números inteiros aleatórios com a função rand(), que aliada ao operador %, gera um valor inteiro entre 0 e NUMERO, sem incluir o NUMERO:

rand() % <NUMERO>

Para utilizarmos essa função sem retornar os mesmos números toda execução, precisamos no início do programa, e somente uma vez, arrumar a semente de geração dos números ao tempo, para isso também precisamos incluir a biblioteca <time.h>:

#include <time.h>

srand(time(NULL));

Então vamos fazer o seguinte, primeiramente uma **função** que solicite ao usuário escolher entre 3 opções:

1. Rolar um dado
2. Rolar mais de um dado
3. Avaliar o problema de Monty Hall (DESAFIO)

Se for escolhida a opção de rolar um dado, deve ser chamada outra **função** de leitura, essa vai ler o número de lados do dado (entre 1 e 100) e o número de repetições de rolagem (maior que 0).

Depois da leitura haverá uma **função** que recebe por parâmetro o número de lados no dado e o número de rolagens, fazer a quantidade de rolagens passadas por parâmetro com um dado com aquela quantidade de lados (utilizar a função rand()). Deverá ser somado cada resultado em um vetor que servirá como contador.

Depois deverá ser chamada uma **função** que recebe esse vetor de contador por parâmetro e o número de rolagens e escreva na tela cada valor e o percentual de vezes que ele caiu (contagem de cada posição / número de rolagens).

Figura 1. Exemplo de rolagens que devem acontecer se o usuário pedir 12 lançamentos de dados de 4 lados

Um	Dois	Três	Quatro
VET[0]	VET[1]	VET[2]	VET[3]
2	4	4	2

```
Distribuição:
1 = 16.66%
2 = 33.33%
3 = 33.33%
4 = 16.66%
```

Figura 2. Exemplo do vetor e da saída da função para os lançamentos da figura anterior

Se for escolhida a opção de rolar mais de um dado, deve ser chamada outra **função** de leitura, essa vai ler o número de dados e o número lados dos dados (a soma total possível de resultado tem que ser menor que 100, por exemplo, 3 dados de 20 lados geram no máximo a soma de 60, então pode, 15 dados de 10 lados somam 150 no máximo, então não pode) e o número de repetições de rolagem (maior que 0).

Depois da leitura haverá uma **função** que recebe por parâmetro o número de dados e de lados nos dados e o número de rolagens, fazer a quantidade de rolagens passadas por parâmetro com aquela quantidade de dados e de lados (utilizar a função rand()). Deverá ser adicionada cada soma de resultados dos dados em um vetor que servirá como contador.

Depois deverá ser chamada uma **função** que recebe esse vetor de contador por parâmetro e o número de rolagens e escreva na tela cada valor e o percentual de vezes que ele caiu (contagem de cada posição / número de rolagens).

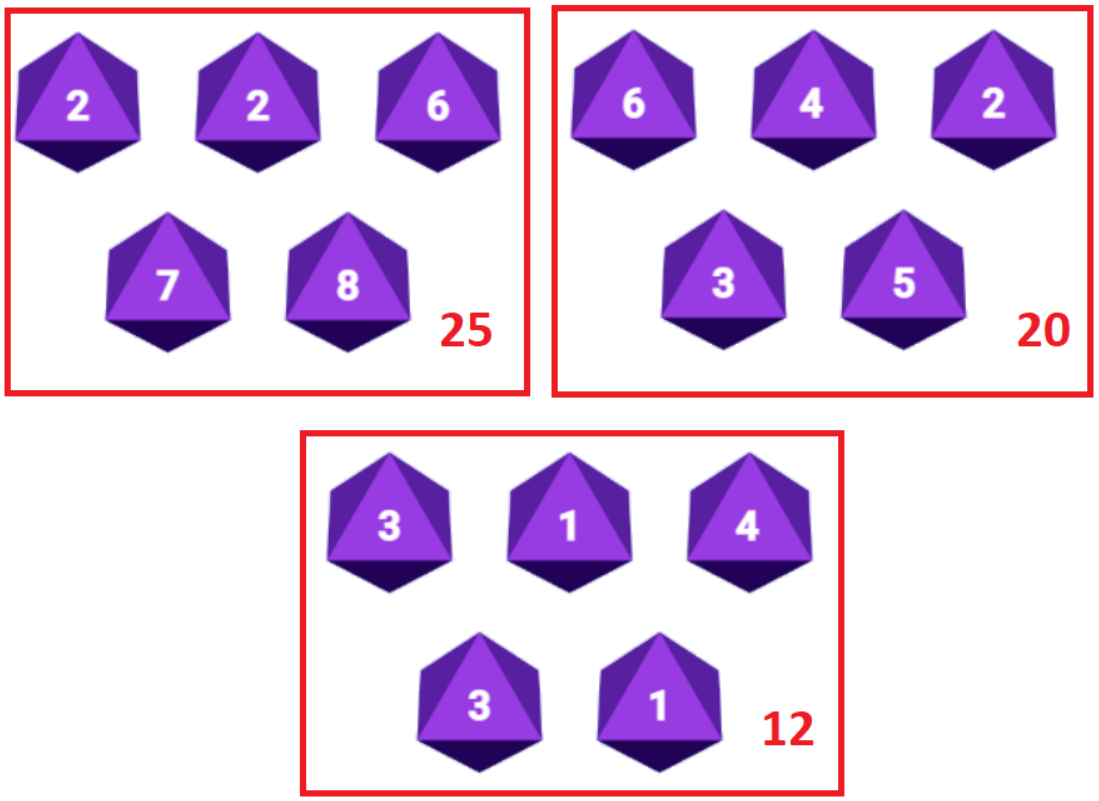

Figura 3. Exemplo de rolagens se a entrada for para 3 lançamentos de 5 dados de 8 lados.

...	Doze	Treze	...	Vinte	...
....	VET[11]	VET[12]	...	VET[19]	...
...	1	0	...	1	...

```
Distribuição:
...
12 = 33.33%
13 = 0%
...
20 = 33.33%
...
```

Figura 4. Exemplo de parte da saída para os resultados acima, lembrando que a saída do programa deve ser completa.

(DESAFIO) Comprove o problema de Monty Hall, faça uma simulação de milhares de jogos com 3 portas e comprove que a mudança de escolha sobe a probabilidade de vitória para 66%. Mais informações sobre o problema em https://pt.wikipedia.org/wiki/Problema_de_Monty_Hall

- As funções de leitura de opções devem passar as variáveis a serem lidas por parâmetro (por referência).
- Para o a guardar os dados de estatística façam um vetor de 100 posições e utilizem até o valor que é necessário, façam uma **função** que reseta todos os valores dele para 0.
- A função de exibir as estatísticas deve ser uma só, chamada pelas funções de rolar um dado ou mais de 1.
- Tanto a função de rolar um ou mais dados deve ter o mesmo nome, elas devem ser sobrecarregadas.
- O problema de Monty Hall é um desafio, não tem peso na nota do trabalho, darei pontos extras para soluções criativas.