FEELT31201 - Programação Procedimental - 2022/1

Laboratório 06

Prazo: 15/01 - 23:59h

Básico 1

1.25 pontos

- Nome do arquivo para o código-fonte: **`inside.c`**
- Tarefa a cumprir:

  Implemente uma função que receba duas strings como parâmetros e verifique se a segunda string ocorre dentro da primeira, retornando o índice de onde começa a primeira (-1 se não contiver). Use aritmética de ponteiros para acessar os caracteres das strings. Peça ao usuário duas strings e retorne se a primeira está contida na segunda; caso afirmativo, mostre onde com auxílio de asteriscos. A função implementada deve obedecer ao seguinte protótipo:

  ```
  int contem(char * s1, char * s2);
  ```

- Exemplo: O usuário entra com `constitucional` e `inconstitucionalissimamente`. Sua função verifica se a primeira está contida na segunda e retorna 2; seu programa então imprime `in*constitucional*issimamente`.
- Método de entrada do usuário:

  ```
  // mensagem para o usuário
  // qualquer entrada de string para interno s1 e para externo s2
  ```

- Casos de teste:

Entrada do usuário	A saída esperada contém
`constitucional inconstitucionalissimamente`	`in*constitucional*issimamente`
`aborigene arborizado`	mensagem que não está contida
`arvore arvoredo`	`*arvore*do`
`mente felizmente`	`feliz*mente*`
`pipa "a pipa do vovo"`	`"a *pipa* do vovo"`
`consequencias sequencia`	mensagem que não está contida

- *Dicas:* Assuma que o usuário não usará diacríticos (acentos e cedilha). Use condicionais dentro do laço que imprime a segunda string para, a partir do resultado da função e o tamanho da primeira string (`strlen` da biblioteca string.h), para imprimir a segunda string com asteriscos, se for o caso.

Básico 2

1.25 pontos

- Nome do arquivo para o código-fonte: `mediaimpares.c`
- Tarefa a cumprir:

  Peça um número N de elementos para um vetor a de inteiros e, em seguida, peça o número inicial s. Se o número inicial for ímpar, considere o predecessor (por exemplo, para 1, considere 0). O programa deve inicializar dinamicamente e preencher o vetor com todos os N números ímpares começando em s. Imprima o penúltimo elemento do vetor a e a média dos números $= \frac{1}{N} \sum_{i=0}^{N-1} a[i]$.

- Exemplo: o usuário deseja informar 5 elementos iniciando com 51 (considerado aqui como 50). A vetor então contém os números 51, 53, 55, 57, 59 imprimindo o penúltimo número 57 e a média $(51 + 53 + 55 + 57 + 59)/5 = 55$.
- Método de entrada do usuário:

```
// mensagem para o usuário
scanf("%d%*c", &n);
// mensagem para o usuário
scanf("%d%*c", &inicial);
```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**	
	penúltimo elemento	média
5 51	57	55
100 1	197	100
1000 42	2039	1042
10000 168	20165	10168
12345678 3	24691355	1.23457e+07

- *Dicas:* Use `double` para sua função média (`%lg` como especificador); o vetor em si pode ser `int`. Você pode reutilizar e adaptar as funções feitas nas questões ou tarefas anteriores. Não se esqueça de liberar sua alocação dinâmica. **Não imprima no terminal todo o vetor na versão final que você disponibilizar para correção!**
- *Sobre especificador no* `scanf`: "Um asterisco opcional (*) logo após o símbolo de porcentagem denota que o dado lido por este especificador de formato não deve ser armazenado em uma variável. Nenhum argumento por trás da string de formato deve ser incluído para esta variável eliminada." (Fonte: `https://en.wikipedia.org/wiki/Scanf_format_string`). Com isso, podemos entender que `%*c` é usado para ignorar o espaço de nova linha (`\n`) ou qualquer outro caractere especial. Portanto, no `scanf()` apresentado aqui, após o usuário inserir o número inteiro, digamos que você terá um caractere de nova linha no buffer que será "engolido" por este `%*c`.

Básico 3

1.25 pontos

- Nome do arquivo para o código-fonte: **`malabarista.c`**
- Tarefa a cumprir:

  Na teoria dos números, uma sequência malabarista é uma sequência inteira que começa com um inteiro positivo a_0, com cada termo subsequente na sequência definida pela relação de recorrência (recursão):

  $$a_{n+1} = \begin{cases} \lfloor \sqrt{a_n} \rfloor & , a_n \text{ par} \\ \lfloor \sqrt{a_n^3} \rfloor & , a_n \text{ ímpar} \end{cases}$$

  Os símbolos matemáticos $\lfloor\ \rfloor$ são equivalentes a `floor` em C (de `math.h`), mais especificamente $\lfloor x \rfloor \equiv$ `(int)floor(x)` ou `(long int)floor(x)`. Ao trabalhar com sequências e vetores, a recorrência pode ser implementada com índices e laços, em vez de recursão, neste trabalho a seu critério.

  Peça ao usuário um número N de elementos para um vetor a de inteiros, então peça o número inicial a_0. O programa deve inicializar dinamicamente e preencher o vetor com todos os números N da sequência do malabarista começando em a_0. Imprima os elementos mínimo e máximo do array a e a média dos números $= \frac{1}{N} \sum_{i=0}^{N-1} a[i]$.

- Exemplo: o usuário deseja informar 5 elementos iniciando com 5. O vetor então contém os números 5, 11, 36, 6, 2, imprimindo o mínimo 2, o máximo 36 e o médio $(5+11+36+6+2)/5 = 12$.
- Método de entrada do usuário:

```
// mensagem para o usuário
scanf("%d%*c", &n);
// mensagem para o usuário
scanf("%d%*c", &a0);
```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**		
	mínimo	máximo	média
5 5	2	36	12
100 21	1	140	4,39
10000 77	1	2322378	264.951
100000 77	1	2322378	27.3951
12345678 77	1	2322378	1.2138

- *Dicas:* `https://en.wikipedia.org/wiki/Juggler_sequence`. Você pode reutilizar e adaptar as funções feitas nas questões ou tarefas anteriores. Como a ordem dos números é a sequência do Juggler, você deve usar `long int` se necessário; ou você pode usar `pow` de `math.h` que retorna um `double`. Use `double` para sua função média (`%lg` como especificador). Não se esqueça de liberar sua alocação dinâmica. **Não imprima no terminal todo o vetor na versão final que você disponibilizar para correção!**

Básico 4

1.25 pontos

- Nome do arquivo para o código-fonte: **`somavetores.c`**
- Tarefa a cumprir:

  Crie uma função para somar dois vetores. Esta função deve receber dois vetores e retornar a soma em um terceiro vetor, garantindo que sejam de tamanhos iguais: caso o tamanho do primeiro e segundo vetor seja diferente, então a função deve retornar `false`; caso a função seja concluída com sucesso, a mesma deve retornar o valor `true` (biblioteca `stdbool.h`). Utilize aritmetica de ponteiros para manipulação do vetor. A função implementada deve obedecer ao seguinte protótipo:

  ```
  bool soma(int * v1, int N1, int * v2, int N2, int * resultado);
  ```

  Peça ao usuário um número N de elementos para um vetor e M para o outro. Preencha o primeiro com ímpares (a partir do 1) e o segundo vetor com múltiplos de 4 (a partir do 0). O programa deve inicializar dinamicamente e preencher o vetor de resposta com N ou M números (se forem iguais) com a soma dos vetores encontrados.

- Exemplo: o usuário deseja usar 7 elementos no primeiro vetor (ímpares, com 7 elementos começando no 1) e 7 no segundo (múltiplos de 4, com 7 elementos começando no 0). Como são do mesmo tamanho, a função retorna `true` e preenche o vetor `resultado` com elementos que são formados pela soma do primeiro e do segundo vetores. No caso, o primeiro será 1, 3, 5, 7, 9, 11, 13 e o segundo será 0, 4, 8, 12, 16, 20, 24. O programa imprime `1, 7, 12, 19, 25, 31, 37`.
- Método de entrada do usuário:

  ```
  // mensagem para o usuário
  scanf("%d%*c", &n);
  // mensagem para o usuário
  scanf("%d%*c", &m);
  ```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**
7 7	`1, 7, 12, 19, 25, 31, 37`
1 2	mensagem que não pode somar vetores
1 1	`1`
10 10	`1, 7, 13, 19, 25, 31, 37, 43, 49, 55`

- *Dicas:* Você pode reutilizar e adaptar as funções que você fez para as questões ou tarefas anteriores. Não se esqueça de liberar sua alocação dinâmica. **Não imprima no terminal todo o vetor na versão final que você disponibilizar para correção!**

Médio 1

1.5 pontos

- Nome do arquivo para o código-fonte: **`raizes2grau.c`**
- Tarefa a cumprir:

  Implemente uma função que calcule as raízes de uma equação do segundo grau do tipo $a\,x^2 + b\,x + c = 0$. Lembrando que:

  $$x = \frac{-b \pm \sqrt{\Delta}}{2\,a}, \qquad \text{onde } \Delta = b^2 - 4\,a\,c$$

  Regras:

  - $a \neq 0$.
  - Se $\Delta < 0$, não existe raiz real.
  - Se $\Delta = 0$, existe **uma raiz** real (duas iguais).
  - Se $\Delta \geq 0$, existem **duas raízes** reais distintas.

  A função implementada deve obedecer ao seguinte protótipo:

  ```
  int raizes(float a, float a, float c, float * x1, float * x2);
  ```

  Essa função deve ter como valor de retorno o número de raízes reais e distintas da equação. Se existirem raízes reais, seus valores devem ser armazenados nas variáveis apontadas por `x1` e `x2`. Peça ao usuário os valores de a, b e c. Seu programa deve mostrar quantas raízes existem e quais são elas (se existem duas, a menor raiz deve ficar em `x1`).

- Exemplo: O usuário informa `1`, `0` e `-1`. O programa então indica que são duas raízes e que são: `-1` e `1`.
- Método de entrada do usuário:

  ```
  // mensagem para o usuário
  scanf("%d %d %d%*c", &a, &b, &c);
  ```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**		
	n. raízes	x1	x2
1 0 1	2	-1	1
1 -1 -12	2	-3	4
5 4 3.7	0		
1.5 1.5 0.375	1	-0.5	
-2 2.5 1	2	-0.3187	1.5687

- **Dicas:** Use conceitos de funções que você desenvolveu para os outros programas. Após a chamada da função, condicione a saída do programa ao retorno do número de raízes.

Médio 2

1.5 pontos

- Nome do arquivo para o código-fonte: `aleatorios.c`
- Tarefa a cumprir:

  Peça ao usuário um número N de elementos para um vetor a de inteiros. O programa deve inicializar dinamicamente e preencher o vetor com N números aleatórios e regras:

  - A semente para o gerador pseudoaleatório também deve ser igual a N (`srand(N)`).
  - Um total de N números são escolhidos aleatoriamente entre 0 e $N-1$ para preencher cada elemento do vetor (`rand%N`);

  Imprima os elementos mínimo e máximo do array a e a média dos números igual a $\frac{1}{N}\sum_{i=0}^{N-1} a[i]$.
- Exemplo: o usuário deseja usar 8 elementos escolhidos aleatoriamente entre 0 e 7 com semente também 8. O array então contém os números 0, 0, 2, 5, 7, 5, 6, 5 imprimindo o mínimo 0, o máximo 7 e a média $(0+0+2+5+7+5+6+5)/8 = 3,75$.
- Método de entrada do usuário:

```
// mensagem para o usuário
scanf("%d%*c", &n); // Também será a semente de aleatoriedade
```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**		
	mínimo	máximo	média
8	0	7	3,75
100	1	97	48,87
10000	0	9998	5018.92
12345678	1	12345677	6.17289e+06

- *Dicas:* Você pode reutilizar e adaptar as funções que você fez para as questões ou tarefas anteriores. Use `srand` apenas uma vez em seu programa, mas depois de saber o valor de N. Use `double` para sua função média (`%lg` como especificador). Não se esqueça de liberar sua alocação dinâmica. **Não imprima no terminal todo o vetor na versão final que você disponibilizar para correção!**
- *Sobre aleatoriedade:* `https://www.javatpoint.com/random-function-in-c`

```
#include <stdlib.h>      // Biblioteca para srand e rand
int main(void) {
    // Usando 42 como semente:
    srand(42);           // Inicialização, deve ser chamada apenas uma vez!
    // ...
    int r = rand();      // Inteiro pseudoaleatório entre 0 e RAND_MAX.
    int s = rand() % 25; // Inteiro pseudoaleatório entre 0 e 24 (inclusos)
    // ...
}
```

Difícil

2.0 points

- Nome do arquivo para o código-fonte: `golomb.c`
- Tarefa a cumprir:

  Para este exercício, vamos utilizar a sequência de Golomb (`https://en.wikipedia.org/wiki/Golomb_sequence`) que é definida por (já ajustada aqui para um índice inicial 0):

  $$a[n] = \begin{cases} 1 & , n = 0 \\ 1 + a[n - a[a[n-1]-1]] & , \text{caso contrário} \end{cases}$$

  Implemente uma função para calcular o elemento da sequência de Golomb. Peça ao usuário o primeiro a e o último índice b a serem considerados (garanta que $b \geq a$). A partir do último índice, inicialize dinamicamente e preencha o vetor com $b+1$ elementos calculados. Mostre o: (i) primeiro elemento de índice a; (ii) o último elemento de índice b; (iii) a média dos números dentro desse intervalo.

- Exemplo: o usuário informa 3 como primeiro índice e 10 como último. O programa inicializa dinamicamente e preenche um vetor de 11 (10+1) elementos com os números 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5. O programa então imprime o primeiro elemento `3`, o último elemento `5` e a média avaliada `4.125` dos números no intervalo desejado do vetor.
- Método de entrada do usuário:

```
// mensagem para o usuário
scanf("%d", &a);
// mensagem para o usuário
scanf("%d", &b);
if(b < a) { aux = a; a = b; b = a; }
```

- Casos de teste:

Entrada do usuário	**A saída esperada contém**		
	primeiro	último	média
0 0	1	1	1
3 10	3	5	4.125
42 12345	12	406	251.803
5225863 5226358	17056	17056	17056
0 10000000	1	25474	15742.4

- **Dicas:** Você pode reutilizar e adaptar as funções que você fez para as perguntas ou tarefas anteriores. Use `double` para sua função média (`%lg` como especificador). Não se esqueça de liberar sua alocação dinâmica e fechar o fluxo de arquivos. **Não imprima no terminal todo o vetor na versão final que você disponibilizar para correção!**