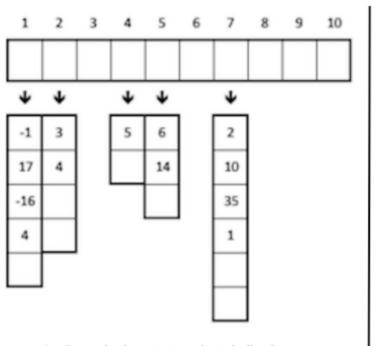


Aluno: _____ Nota: _____

Avaliação 3 – 2024.2

1) Considere, como exemplo, a Figura 1.



A - Exemplo da estrutura do trabalho 2

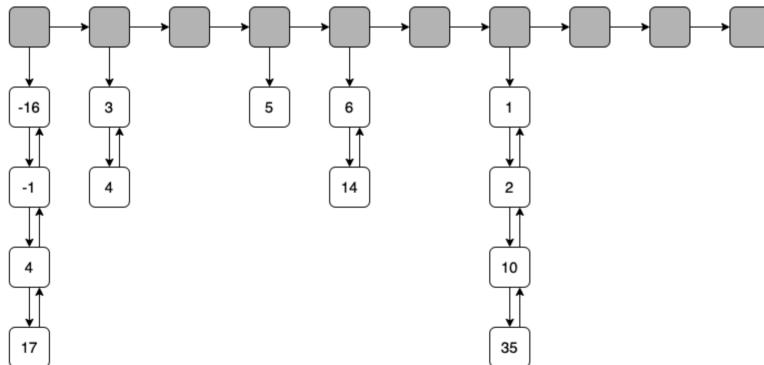
Trabalho3 > dadas.txt

```

1 1;5;-1;17;-16;4
2 2;4;3;4
3 4;2;5
4 5;3;6;14
5 7;6;2;10;35;1

```

B - Exemplo de arquivo criado, com os dados do item A



C - Estrutura de dados criada a partir dos dados exemplo do arquivo do item B

Figura 1: Exemplos das estruturas

- A Figura 1-A possui um exemplo de estrutura utilizada no trabalho 2;
- A Figura 1-B tem um exemplo de arquivo, criado por um programador, para armazenar os dados da estrutura da Figura 1-A. Em cada linha foi representado as informações de um vetor auxiliar, sendo: a posição referente à estrutura principal, o tamanho da estrutura auxiliar, e os elementos que existiam em um determinado momento.
- A Figura 1-C possui uma nova estrutura de dados que lembra a estrutura da Figura 1-A. Porém ela tem sua lista principal implementada como uma lista encadeada. E as estruturas auxiliares foram implementadas com uma lista duplamente encadeada ordenada.
- Ao finalizar o programa, deve ser possível, a partir da estrutura da Figura 1-C, gerar um arquivo como o da Figura 1-B

Faça o que se pede a seguir:

A) (Valor 0.5) Defina, no espaço em branco a seguir, uma embaixo da outra, as estruturas de dados necessárias para representar as estruturas da Figura 1-C, e para inicializar o código. Além do código, coloque ao lado, um texto explicativo dizendo a que se refere cada estrutura que foi criada.

B) (Valor 1.5) Crie uma função recursiva **lerArquivo**. Essa função deve ser capaz de ler o arquivo exemplo da Figura 1-B, e retornar os dados para quem chamar ela. Não pode haver aqui parâmetro global.

C) Crie uma função **gerarEstruturaNova**, que recebe todos os dados lidos na função **lerArquivo**, e monte a estrutura correspondente da Figura 1-C.

D) Crie a função **main**, que defina as estruturas necessárias para o programa funcionar, que chame as funções **lerArquivo** e **gerarEstruturaNova** passando e recebendo os parâmetros necessários.

Observação: Não precisa implementar o que é dito no texto a seguir, mas a main deve ter a capacidade, caso necessário, de enviar os dados da estrutura representada pela Figura 1-C para que seja salva num arquivo no formato da Figura 1-B.

2) Considere o código abaixo

```
1 #include <stdio.h>
2
3 void fb(int *a){
4     *a = *a + 1;
5     if (*a > 1)
6         printf("a1: %d\n", *a);
7     printf("a2: %d\n", *a);
8 }
9
10 void fa(int *b){
11     int *c = b;
12     int *d;
13     if (*c > 0){
14         *c = *c - 1;
15         fa(c);
16         *b = *b - 1;
17         fa(b);
18         d = c;
19         if (*d % 2 != 0){
20             fb(d);
21             printf("d: %d\n", *d);
22         }
23         *d = *d - 1;
24         fa(d);
25         printf("c: %d\n", *c);
26         printf("b: %d\n", *b);
27     }
28 }
29
30 int main(){
31     int i = 3;
32     int *a = &i;
33     fa(a);
34 }
```

a) (Valor 3.0) Ao compilar e executar o código da figura ao lado, qual a saída do programa? Responda essa questão no espaço abaixo.

Observações gerais

- leia a prova cuidadosamente. Caso não tenha entendido o que se pede, ou caso visualize alguma ambiguidade, informe o professor, para que ele possa esclarecer suas dúvidas
- escreva seu código de forma legível
- implemente apenas o que se pede
- coloque seu nome em todas as folhas utilizadas