

Exercício para Nota

23/09/2023 – período manhã

Ao entregar esta atividade no Teams, será garantida a presença para o dia 23/09/2023 no período da manhã.

Estacionamento

Um estacionamento possui uma única entrada/saída e a capacidade máxima para guardar dez carros. Faça um programa que simule a entrada (empilhar) e saída (desempilhar) de carros deste estacionamento. Armazene a placa e ano de cada carro estacionado. A entrada de dados deve ocorrer dentro da função de empilhamento.

Os protótipos das funções estão definidas.

As estruturas de dados estão definidas como o modelo apresentado:

```
struct Carro {
    string placa;
    int ano;
};

struct Pilha {
    Carro data[TAMANHO_PILHA];
    int size;
};
```

- 1) Desenvolver a função inicializar. (**void inicializar(Pilha *)**;). Esta função atribui o valor zero para o contador. Também deve percorrer o vetor de carros e atribuir “AAA0000” para todas as placas e o valor “0” para todos os anos. **(1 ponto)**
- 2) Desenvolver a função estacionar (**void estacionar(Pilha *, string, int)**;). Esta função recebe a placa do carro no segundo argumento e o ano do carro no terceiro argumento. A função deve empilhar o carro na pilha e incrementar o contador. **(2 pontos)**
- 3) Desenvolver a função estacionar (**void estacionar(Pilha *, Carro)**;). Esta função recebe o carro no segundo argumento . A função deve empilhar o carro na pilha e incrementar o contador. **(2 pontos)**
- 4) Desenvolver a função sair(**Carro sair(Pilha *)**;). Esta função retorna o carro no topo da pilha. A função deve atribuir o valor “XXX0000” a placa e o valor “0” ao ano do carro desempilhado. **(2 pontos)**
- 5) Desenvolver a função imprime_pilha (**void imprime_pilha(Pilha *)**;). Esta recebe um ponteiro da estrutura Pilha e imprime somente os carros empilhados. **(1 ponto)**
- 6) Desenvolver a função imprime_tudo (**void imprime_tudo(Pilha *)**;). Esta recebe um ponteiro da estrutura Pilha e imprime todos os elementos da pilha, inclusive os carros preenchidos com o valor padrão. **(1 ponto)**
- 6) Desenvolver a função quantidade_vagas (**int quantidade_vagas(Pilha)**;). Esta recebe a estrutura Pilha retorna a quantidade de vagas disponíveis no estacionamento. **(1 ponto)**
- 7) Descomentar os comentários entre as linhas 37 e 55. Ao descomentar, compilar e executar, espera-se que a saída do programa seja semelhante ao apresentado na próxima página. **(0 pontos)**

Saída esperada ao descomentar entre as linhas 37 e 55 (função main)

Quantidade de vagas 8
PILHA DE CARROS

=====

DOW5782 2004
ABC1234 1999

PILHA DE CARROS

=====

HIX5215 2008
GIN3285 2017
WQT2245 2020
XYZ3215 2022
DOW5782 2004
ABC1234 1999

Saiu o carro: HIX5215

PILHA DE CARROS

=====

GIN3285 2017
WQT2245 2020
XYZ3215 2022
DOW5782 2004
ABC1234 1999

Quantidade de vagas 5
TODA A PILHA

=====

AAA0000 0
AAA0000 0
AAA0000 0
AAA0000 0
XXX0000 0
GIN3285 2017
WQT2245 2020
XYZ3215 2022
DOW5782 2004
ABC1234 1999