Universidade Federal de Goiás
Instituto de Informática
Introdução à Programação
Prova4 - Structs e Alocação Dinâmica de Memória

Prof. Msc. Elias Batista Ferreira
Prof. Msc. Gilmar Ferreira Arantes
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

19/11/2020

Instruções para a Resolução dos Problemas

Esta avaliação é composta por 3 questões, variando do nível de dificuldade 1 a 4. O conteúdo abordado envolve, structs e alocação dinâmica, sendo uma questão de nível 1, uma de nível 2 e uma de nível 4.

Você deve resolvê-la seguindo as seguintes orientações:

1. A entrega desta prova é através do sistema Sharif.
2. Você deve implementar cada uma das questões no seu próprio computador e submeter ao Sharif, na área do problema correspondente à questão da prova.
3. A data e horário para entrega é 19/11/2020 - 23h59min.
4. A detecção de plágio (código duplicado) define nota 0 (zero) para os alunos envolvidos.
5. O professor estará disponível para esclarecimento de dúvidas. Sendo que até as 20hs estará *online*, das 20hs até as 22hs estará *offline*. Após as 22hs não haverá mais esclarecimento de dúvidas.

Sumário

1 Strings Econômicas (+)

 (+)

Escreva um programa em C para ler n nomes de pessoas na entrada e armazenar esses nomes em um vetor em que cada elemento deve ter espaço suficiente apenas para armazenar um nome lido e o caractere '\0'. Cada nome deve ser lido primeiramente em um *buffer* que é uma string com memória suficiente para armazenar mais do que o maior nome esperado (ex. 10000 caracteres). Em seguida deve ser alocado espaço num elemento do vetor que seja suficiente apenas para armazenar a string que está no buffer mais o caractere delimitador de cadeia. Em seguida, o programa copia (com strcpy) a cadeia no buffer para o espaço alocado no vetor e volta a ler outra cadeia no buffer. O programa deve imprimir todas as strings no vetor e, antes de terminar, deve liberar todo espaço alocado dinamicamente no vetor.

Observação

Você pode usar as funções *malloc* e *free* da stdlib para alocar e liberar espaço para armazenar os elementos do vetor. Pode usar a função *strcpy* para copiar uma string lida no buffer para um elemento alocado no vetor.

Entrada

A primeira linha da entrada contém o número n ($1 \leq n \leq 30$) de nomes a serem lidos. Em seguida, aparecem n linhas, cada uma com um nome.

Saída

A saída é formada por n linhas cada uma contendo um nome armazenado no vetor.

Exemplo

Entrada
3 Joao Antonio Maria Pedro Rezende de Souza Adriana Lucia de Assis
Saída
Joao Antonio Maria Pedro Rezende de Souza Adriana Lucia de Assis

2 Estatística Empresarial (++)

 (++)

Uma empresa deseja obter informações estatísticas sobre seus funcionários. Para isso, irá colher os seguintes dados dos funcionários: matricula, idade, número de filhos, sexo e salário.

Essas informações precisam ser armazenadas em uma estrutura:

```
typedef struct {
    int matricula;
    int idade;
    int numFilhos;
    char sexo;
    double salario;
} FUNCIONARIO;
```

Entrada

Na primeira linha ha um inteiro n, 1<n≤500, representando a quantidade de funcionário (fazer alocação dinâmica). A seguir haverá n linhas com n os seguintes dados separados por um espaço em branco cada: matricula, idade, número de filhos, sexo e salário.

Saída

Deverá imprimir 4 (quatro) respostas:

- Quantidade de funcionários com idade superior a média de idades E salário superior a 3 salários mínimos.
- Quantidade de mulheres que possuem quantidade de filhos acima da média geral.
- Quantidade de homens que possuem quantidade de filhos acima da média geral.
- Quantidade de funcionários maiores de 47 anos com renda per-capita (por pessoa) abaixo de 2 salários mínimos.

* considere o salário mínimo igual a 1200.00.

Exemplo

Entrada	Saída
10 101 44 4 M 7001.00 105 56 2 F 2950.00 211 60 2 F 6870.00 221 25 1 F 9200.00 231 38 3 M 4350.00 300 70 4 M 2100.00 545 27 0 F 4500.00 654 65 1 F 2900.00 670 53 2 M 3300.00 888 55 2 F 4100.00	2 0 3 6

3 Loteria (++++)

 (++++)

A Loteria é um jogo que paga um prêmio em dinheiro para o apostador que conseguir acertar os 6 números sorteados. Ainda é possível ganhar prêmios ao acertar 4 ou 5 números dentre os 60 disponíveis no volante de apostas. Para isso, você deve *marcar* 6 números do **volante**. Você poderá fazer quantas apostas quiser, ou seja, poderá jogar quantos volantes necessitar. Os números estão entre 1 e 60.

Faça um programa que receba os jogos de um apostador, em seguida, leia o resultado da loteria e verifique se o apostador acertou os números sorteados. Se o apostador acertou 4, 5 ou 6 números é necessário emitir um aviso reportando o fato.

É obrigatório utilizar estrutura para armazenar os números apostados e o resultado .

```
typedef struct {
    int numJogo;
    int numero[6];
} CARTELA;
```

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro $N(1 \leq N \leq 10^3)$, indicando a quantidade de apostas do jogador. As N linhas seguintes contém o número do jogo e 6 números correspondentes aos palpite do jogador.

Em seguida, deverá ter um linha para ler o número do concurso e os 6 números sorteados, que devem ser armazenados em outra estrutura.

*** Deve-se utilizar alocação dinâmica para reservar N espaços das apostas.**

Saída

Para cada entrada, deve-se verificar se o apostador acertou, no mínimo, 4 números e emitir a seguinte mensagem:

1. QUADRA jogo: a b c d: quando a apostador acertar 4 números.
2. QUINA jogo: a b c d e: quando a apostador acertar 5 números.
3. SENA jogo: a b c d e f: quando a apostador acertar 6 números.

Após analisar todas as apostas e constatar que o apostador não conseguiu acertar, no mínimo, 4 números, escreva a mensagem "NENHUMA CARTELA PREMIADA PARA O CONCURSO concurso".

Exemplos

Entrada
4 1 5 15 25 35 45 55 2 9 13 28 46 51 52 3 2 28 46 47 51 13 4 8 15 25 35 45 55 1050 9 13 28 46 51 52
Saída
SENA 2: 9 13 28 46 51 52 QUADRA 3: 28 46 51 13

Entrada
3 1 3 11 44 50 56 32 2 2 12 57 51 45 33 3 1 34 13 46 58 52 1051 5 15 36 47 53 60
Saída
NENHUMA CARTELA PREMIADA PARA O CONCURSO 1051