

COMPUTAÇÃO ELETRÔNICA - IF165

LISTA 6 - STRUCTS E PONTEIROS

1 - SENHA DA EMPRESA CALCULAÍ (3640)

Uma pequena empresa, com 5 funcionários, chamada Calculaí, está em busca de criar uma nova forma de identificar seus funcionários. Assim, o seu objetivo é fazer um programa que receba uma **estrutura** com os campos correspondentes ao **NOME COMPLETO**, **DATA DE NASCIMENTO**, **CIDADE NATAL** e **IDADE** de um funcionário dessa empresa e retorne a sua senha de identificação.

Para a formação da senha, você, estudante de Computação Eletrônica, deverá concatenar alternadamente a primeira letra do nome, o primeiro número da data de aniversário, a primeira letra da cidade natal e assim sucessivamente, até atingir **as cinco primeiras letras do nome, o dia e mês do aniversário, as quatro primeiras letras do nome da cidade natal** e, para diminuir a possibilidade de haver repetições, terminar com a **idade**. Ou seja, se a entrada for dada por **NomeC, 28/12, Cidade** e **77**, a senha gerada deverá ser **N2Co8im1de2aC77**.

Formato de entrada

O programa receberá de cada um dos funcionários os campos correspondentes a estrutura, com:

- Uma string com, no máximo, 50 caracteres para o NOME COMPLETO.
- Uma string para a DATA DE NASCIMENTO no formato DIA/MÊS/ANO.
- Uma string com, no máximo, 50 caracteres para a CIDADE NATAL.
- Um inteiro para a IDADE.

Os dados serão computados até que o NOME COMPLETO seja "SAIR" ou a quantidade de funcionários da empresa seja atingida.

Observação: o primeiro nome do NOME COMPLETO sempre terá no mínimo cinco letras.

Formato de saída

A saída deverá seguir o modelo, com uma quebra de linha dupla após ser printado a senha do funcionário:

Nome: <nome_completo>

Data de Nascimento: <XX/XX/XXXX>

Cidade Natal: <cidade_natal>

Idade: <idade>

Cadastro: <senha_gerada>

Exemplo de entrada

Damiao dos Santos Silva

24/07/2000

Recife-PE

21
SAIR

Exemplo de saída

Nome: Damiao dos Santos Silva
Data de Nascimento: 24/07/2000
Cidade Natal: Recife-PE
Idade: 21 anos
Senha: D2Ra4em0ci7ia21

2 - GABI ADORA DESAFIOS (3620)

Gabi entrou na faculdade de Engenharia e está muito animada com a cadeira de Cálculo 1, por isso ela quer usar seu conhecimento de linguagem C para criar um programa que calcule a derivada de uma função polinomial, ou seja:

$$\frac{d3x^4}{dx} = 3 \cdot 4x^{4-1} = 12x^3$$

Gabi irá inserir quantos monômios terá a função e depois irá inserir um monômio de cada vez. Assim, considerando a função $f(x) = 3x^3 - 2x + 3$, teríamos o modelo de entrada:

3
3 x 3
-2 x 1
3 0 0

Observe que, no caso de uma constante, a incógnita e o expoente são iguais a 0. Dessa forma, você deve criar uma função com assinatura ***void derivada (int *coeficiente, int *expoente)*** que receba como parâmetro o endereço do coeficiente e do índice de cada monômio e calcule a derivada. Note que os valores devem ser passados por referência, com ponteiros, e não por cópia. A sua função principal deve imprimir a derivada seguindo o formato de saída até que o número de monômios inseridos seja igual a 0.

Além disso, desconsidere casos com expoentes negativos.

Lembre-se que:

1. Gabi pode inserir outras incógnitas além de x;
2. A derivada de uma constante é 0;
3. $x^0 = 1$, mas o 1 não deve ser impresso;

Formato de entrada

A quantidade de monômios, seguido pelos monômios.

3
3 x 3
-2 x 1
3 0 0

0

Formato de saída

A derivada da função seguindo o formato: $f'(x)=+9x^2-2$

Exemplo de entrada
3
3 x 3
2 x 1
3 0 0
3
2 x 4
2 x 3
2 x 2
0
Exemplo de saída
$f'(x)=+9x^2 + 2$
$f'(x) = +8x^3 + 6x^2 + 4x^1$

3 - ANALISANDO VENDEDORES (3623)

Ricardo tem várias lojas de artigos esportivos e precisa de você para desenvolver um programa que o ajude a registrar e analisar o rendimento dos vendedores. Esse programa deve indicar o melhor vendedor, que receberá um bônus, o pior vendedor, que ficará em acompanhamento e também permitir que Ricardo pesquise diretamente pelo nome do vendedor as informações relacionadas ao mesmo (nome, vendas, unidade que trabalha e gerente responsável).

Para executar essa tarefa, você deverá construir um programa com duas funções além da principal, as quais possuirão as seguintes assinaturas:

- ***void buscarMelhorPior(Vendedor vetor[], int tamanho, Vendedor melhorpior[])***: deverá procurar no vetor principal qual o melhor e pior vendedor (relacionados aos valores das vendas realizadas no período) e armazenar todas as informações referentes aos mesmos numa nova estrutura, contendo apenas as informações do melhor e pior.
- ***void buscarVendedor(char nome[], Vendedor vetor[], int tamanho)***: deverá buscar na estrutura o vendedor solicitado e imprimir na tela as informações do mesmo, até que o nome a ser pesquisado seja “ok”. Caso não seja encontrado, imprimir “NAO ENCONTRADO”.

O programa principal ***main()*** deverá realizar as seguintes operações:

1. Solicitar do usuário a quantidade de vendedores a serem cadastrados.
2. Ler os dados dos vendedores, os quais deverão ser inseridos pelo usuário, e preencher o vetor de estruturas de vendedores.
3. Chamar a função ***void buscarMelhorPior(Vendedor vetor[], int tamanho, Vendedor melhorpior[])***, passando como parâmetros o vetor de estrutura de cadastro, a quantidade de vendedores e a estrutura melhorpior.

4. Chamar a função ***void buscarVendedor(char nome[], Vendedor vetor[], int tamanho)***, passando como parâmetros o nome do vendedor a ser pesquisado, o vetor de estrutura de cadastro e a quantidade de vendedores.
5. Encerrar o programa.

Formato de entrada

O formato de entrada será dado por:

- Quantidade de vendedores cadastrados: um inteiro n , com $0 < n < 50$.
- Quantidade de vendedores cadastrados: um inteiro n , com $0 < n < 50$.
 - Nome do vendedor: string com no máximo 50 caracteres;
 - Valor das vendas realizadas no período: float;
 - Unidade de trabalho: inteiro;
 - Gerente responsável: string com no máximo 50 caracteres;
- Nome dos vendedores que deverão ser pesquisados.

Formato de saída

O formato de saída fornecerá as informações dos vendedores separadas por quebra de linha e as informações do melhor e pior, assim como dos vendedores pesquisados separadas com uma quebra de linha dupla, seguindo o modelo abaixo:

Melhor vendedor:

Vendas: R\$

Unidade:

Gerente:

Pior vendedor:

Vendas: R\$

Unidade:

Gerente:

Nome:

Vendas: R\$

Unidade:

Gerente:

NAO ENCONTRADO

Exemplo de entrada

5
Alberto Ferreira
1230.67
4
Josué Carlos
João da Silva
1720.00
4
Josué Carlos
Sheila Carvalho
3465.00
3
Ingrid Correia
Carlos Duarte
3296.00
3
Jonas Santos
Julia Ferreira
1002.12

2
Filipe Costa
Carlos Duarte
Lucas Nilo
Julia Ferreira
ok

Exemplo de saída

Melhor vendedor: Sheila Carvalho
Vendas: R\$ 3465.00
Unidade: 3
Gerente: Ingrid Correia

Pior vendedor: Julia Ferreira
Vendas: R\$ 1002.12
Unidade: 2
Gerente: Filipe Costa

Nome: Carlos Duarte
Vendas: R\$ 3296.00
Unidade: 3
Gerente: Jonas Santos

NAO ENCONTRADO

Nome: Julia Ferreira
Vendas: R\$ 1002.12
Unidade: 2
Gerente: Filipe Costa

4 - LISTA DE CONTATOS (3611)

Joãozinho está criando uma lista de contatos e precisa de sua ajuda. Ele precisa que para cada contato você crie uma estrutura com os seguintes itens:

- Nome e sobrenome;
- Número no formato (00)01234-5678;
- Código informando a relação que João tem com a pessoa, sendo:
 - 1 para família;
 - 2 para faculdade;
 - 3 para amigo;

Você deve criar um programa que receba essas informações e também permita que Joãozinho pesquise os contatos, através do nome, retornando as informações referentes ao mesmo, até que o usuário digite "". Caso Joãozinho não tenha cadastrado essa pessoa, imprima um aviso: "<nome pesquisado> nao foi cadastrado".

Note que Joãozinho irá pesquisar apenas pelo primeiro nome, então caso existam duas pessoas com o mesmo primeiro nome, imprima as duas.

Exemplo:

Entrada:

Maria Eduarda

(00)01234-5678

1

Maria Julia

(00)01234-5678

2

Saída:

Se Joãozinho pesquisar por Maria, você deve imprimir:

Nome: Maria Eduarda

Numero: (00)01234-5678

Vinculo: Familia

Nome: Maria Julia

Numero: (00)01234-5678

Vinculo: Faculdade

Formato de entrada

O formato de entrada consiste em:

1. Um inteiro n indicando quantos contatos serão adicionados ($0 < n < 50$);
2. As informações dos contatos na ordem:
 - Nome: string de no máximo 50 caracteres;
 - Número: string no formato (00)01234-5678;
 - Inteiro indicando o código de vínculo;
3. Os nomes a serem pesquisados, até o usuário digitar "#".

Formato de saída

O formato de saída consiste nas informações abaixo separadas por quebra de linha dupla:

Nome: Nome Sobrenome

Numero: (00)01234-5678

Vínculo: Família, Faculdade ou Amigo (de acordo com o código de vínculo)

Observe que, os nomes devem ser impressos na ordem em que foram pesquisados e caso o nome pesquisado tenha mais de um contato, imprimir na ordem em que foi recebido. Por fim, se o nome pesquisado não estiver cadastrado, imprimir: "<nome pesquisado> nao foi cadastrado".

Exemplo de entrada

```
6
Renata Santos
(81)98989-7876
2
Roberto Elias
(81)98659-7236
1
Roberto Ferreira
(81)98959-7236
3
Gabriela Fontes
(81)98959-3454
2
Joao Silva
(81)99273-3454
1
Carlos Dantas
(81)99273-2342
3
Roberto
Joao
Julia
```

Exemplo de saída

Nome: Roberto Elias
Numero: (81)98659-7236
Vinculo: Familia

Nome: Roberto Ferreira
Numero: (81)98959-7236
Vinculo: Amigo

Nome: Joao Silva
Numero: (81)99273-3454
Vinculo: Familia

Julia nao foi cadastrado