

Prova 2

Programação e Desenvolvimento de Software I

Professor: Pedro O.S. Vaz de Melo

Nome: _____

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Função/Operador	Descrição	Exemplo
<code>rand()</code>	gera um número aleatório inteiro entre 0 e <code>RAND_MAX</code>	<code>rand()</code> pode gerar 41
<code>RAND_MAX</code>	o maior número possível que pode ser gerado por <code>rand()</code>	<code>RAND_MAX</code> é sempre 32767

1. (4 points) Escreva uma função de nome `randFloat` que retorna um número ponto flutuante aleatório entre um valor mínimo (`min`) e máximo (`max`). A sua função deve ser capaz de gerar números aleatórios com parte fracionária diferente de zero e deve ter o seguinte protótipo:

```
float randFloat(float min, float max);
```

Importante: nas questões a seguir, você vai implementar funcionalidades de um sistema de dieta. O sistema tem um registro de 100 alimentos e de suas propriedades. Além disso, o sistema avaliará o consumo calórico de dietas de usuários e indicará qual alimento é o mais popular. Use a definição abaixo sempre que necessário:

```
#define NUMALI 100
```

2. (2 points) Defina um **novo tipo de dados** para representar um **Alimento**, que deve ser capaz de armazenar as seguintes informações: quantidade de carboidratos (g), proteínas (g), gordura (g) e calorias (kcal), todos esses pontos flutuantes.

3. (4 points) Implemente uma função de nome `criaAlimento` que recebe um **Alimento por referência** e preenche os seus campos da seguinte forma. A quantidade de carboidratos é um número aleatório entre 10.5 e 25, a quantidade de proteínas é um número aleatório entre 0 e 15.5 e a quantidade de gordura é um número aleatório entre 0 e 10.5. A quantidade de calorias é dada pela seguinte maneira. Para cada grama de carboidrato e proteína, some 4 kcal, e para cada grama gordura, some 9 kcal.

4. (4 points) Implemente uma função de nome `criaMenu` que recebe como parâmetros um vetor de **Alimentos** e a quantidade de alimentos suportados pelo sistema. A função deve preencher cada uma das posições desse vetor com um **Alimento** criado pelo função `criaAlimento`. Protótipo:

```
void criaMenu(Alimento menu[], int n);
```

5. (4 points) Implemente uma função de nome `criaDieta` que recebe como parâmetros um vetor de inteiros `dieta` e a quantidade de alimentos cadastrados no sistema `n`. A posição `i` do vetor `dieta` indica se o alimento de índice `i` está presente na dieta (`dieta[i]>0`) e em quantas porções. Você deve preencher cada posição do vetor de forma aleatória e da seguinte maneira. Para cada alimento `i`, gere um número aleatório `r` entre 0 e 1 para verificar se o alimento está na dieta. Caso `r < 0.3`, o alimento está na dieta e você deve sortear uma quantidade de porções `k`, que deve ser um número aleatório inteiro entre 1 e 5. Depois, assinala `dieta[i] = k`. Se o alimento não está na dieta, então `dieta[i] = 0`. Protótipo:

```
void criaDieta(int dieta[], int n)
```

6. (4 points) Implemente uma função de nome `totalCalorias` que recebe como parâmetros um vetor de inteiros correspondente a uma dieta, o menu de Alimentos do sistema, e a quantidade de alimentos do sistema n . A função deve retornar a quantidade total de calorias dessa dieta. Protótipo:

```
float totalCalorias(int dieta[], Alimento menu[], int n);
```

7. (6 points) Implemente uma função de nome `AlimentoMaisPop` que recebe como parâmetro um número m de pessoas e o vetor de dietas dessas m pessoas representados em uma matriz $m \times n$, em que n é a quantidade de alimentos do sistema, ou simplesmente `NUMALI`. A função deve retornar o índice do alimento mais presente nas dietas dessas pessoas. Desconsidere empates. Protótipo:

```
int AlimentoMaisPop(int dietas[][NUMALI], int m);
```

Exemplo: a matriz abaixo descreve a dieta de 4 pessoas de um sistema com 5 alimentos cadastrados, isto é, `NUMALI = 5`. Para essa matriz, a sua função deve retornar 2, pois esse alimento ocorre em três dietas, mais que todos os outros alimentos.

0	0	0	1	2
0	4	2	0	0
0	0	1	0	4
3	3	1	1	0

8. (4 points)

Complete o código abaixo:

```
void main() {
    Alimento menu[NUMALI];

    criaMenu(_____);

    //cria dietas para 10 pessoas:
    int i, diets[10][NUMALI];
    for(i=0; i<10; i++) {

        criaDieta(_____);
        //qtd de calorias:
        printf("\nkcal de %d: ", i);

        printf("%f\n", _____);
    }

    printf("\nalimento mais usado:");

    printf("%d\n", _____);
}
```