Escola Superior de Tecnologia e Gestão de Viseu

Engenharia Informática

Exame - Época Normal **Algoritmos e Programação**

Ano letivo 2022/2023 – 1º Ano/1º Semestre

25.01.2023 **Duração:** 120 Minutos Sem Consulta

engenharia informática

Nome: _____ *N.º* | | | | | | |

Notas: Neste teste deve considerar que todas as secções de código pedidas têm de ser escritas na linguagem de programação C.
Todo o material fornecido pelo docente deve ser entregue no final da prova.
Não é permitida a utilização de qualquer dispositivo eletrónico.

1. (2.0V)

a) Descreva o que faz o programa:

```
#include <stdio.h>
int funcao(int *m)
{
  m++;
  (*m)++;
  return *m;
}
void main(){
  int v[4]={3,4,6,10};
  printf("\n%d",funcao(v));
}
```

b) Considere a estrutura:

```
typedef struct ponto{
   int x,y;
}PONTO;
```

e o excerto de código:

```
PONTO *p,ponto={0,0};
p = &ponto;
printf("(%d, %d)\t", (*p).x, p->y);
ponto.x = 10;
p->y = 20;
printf("(%d, %d)\n", p->x, ponto.y);
```

O excerto de código apresenta erros? Em caso negativo, escreva o que é mostrado na consola.

2. (2.0V) Elabore uma função que, dada uma string, devolve outra, mas sem as vogais.

```
char * stringSemVogais(char * str)
```

3. (2.0V) Elabore uma função que, dado um vetor com N elemento inteiros, calcule e devolva a maior diferença existente entre os seus elementos.

```
// @param v: Vetor
// @param N: Número de elementos do vetor anterior
int maiorDiferenca(int v[],int N)
```

4. Considere o seguinte excerto de código:

```
#define MAX 12

typedef struct atleta{
   char nome[30];
   char pos[8];     //posição do atleta
   float mPontos;   //média pontos
   float mPerdas;   //média perdas
   float mAssist;   //média assistências
   float mRessalt; //média ressaltos
   int  tempo;      //tempo de jogo
}ATLETA;
```

```
typedef struct equipa {
  char nome[50];
  ATLETA atletas[MAX];
}EQUIPA;
```

Suponha que a valia de um atleta de basquetebol é calculada de acordo com a sua <u>posição</u> (Base, Extremo ou Poste) segundo as fórmulas:

Base: valia = 3*mPontos + 3*mAssist + 1*mRessalt – 3*mPerdas

Extremo: valia = 4*mPontos + 2*mAssist + 2*mRessalt – 3*mPerdas

Poste: valia = 3*mPontos + 1*mAssist + 3*mRessalt – 3*mPerdas

a) (2.5V) Implemente uma função que devolva a valia de uma equipa.

```
// @param equipa: equipa para a qual se pretende determinar a valia
float valiaEquipa(Equipa equipa)
```

b) (2.5V) Desenvolva uma função que devolva a valia e o nome da equipa menos valiosa. Deve usar a função anterior para calcular a valia de uma equipa.

```
// @param equipas: Vetor de equipas
// @param n: Número de equipas no vetor anterior
// @param nomeEquipa: Nome da equipa menos valiosa
float equipaMenosValiosa(Equipa *equipas, int n, char *nomeEquipa)
```

c) (2.5V) Elabore uma função que escreva o nome e a valia do extremo mais valioso de cada equipa. Para esse efeito, só devem ser considerados extremos com mais de 100 minutos de jogo.

```
// @param equipas: Vetor de equipas
// @param n: Número de equipas no vetor anterior
void extremoMaisValioso(Equipa *equipas, int n)
```

d) (2.5V) Desenvolva uma função que que escreva, num ficheiro de texto, os nomes, posições e os tempos de jogo dos atletas de uma determinada equipa.

```
// @param equipas: Vetor de equipas
// @param n: Número de equipas no vetor anterior
// @param nomeEquipa: Nome da equipa
// @param f: ficheiro a gravar
void gravarEquipa(Equipa *equipas, int n, char *nomeEquipa, FILE *f)
```

5. (2.0V) Considere a função aux. Explique o funcionamento da função quando se chama aux(3,L,0,5), em que L={-2,-2,1,3,5,6,7,8}.

```
int aux(int x, int L[ ],int inic, int fim){
 int meio = (inic+fim)/2;
 if (inic > fim)
  return 0;
 else {
  if (x == L[meio])
    return 1;
  else{
    if (x > L[meio])
      return aux(x, L, meio+1, fim);
    else
      return aux(x, L, inic, meio-1);
    }
  }
}
```

6. (2.0V) Elabore uma função que calcule, de forma eficiente, o valor do somatório $\sum_{i=1}^{N} \frac{i!}{3^{i-1}}$