Escola Superior de Tecnologia e Gestão de Viseu

Engenharia Informática

PROVA MODELO de **Algoritmos e Programação**

Ano letivo 2023/2024 – 1º Ano/1º Semestre

Duração: 120 Minutos Sem Consulta

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 a função se v[6]={2,1,2,2,1,-4} entrarem na função:

```
int semNome (int v[], int n)//n é dimensão do vetor
{
    int i;v=v+n-1;
    for ( i=n-1; i > 0; i--){
       if(*v==*(v-1)){
        printf("\n%d %d\n", *v, *(v-1));
        return 0;}
       v--;}
    return *v;}
```

b) Considere a estrutura:

```
typedef struct entrada
{
  char nome[20];
  float preco[2];
} ENTRADA;
ENTRADA lista[20];
```

Escreva as instruções para imprimir o 2º caracter do nome e a média de preços, referentes ao 10º elemento da lista.

2. (2.5V) Considere a estrutura

```
typedef struct complex
{
    float real;
    float imag;
} complex;
```

Elabore a função, com a assinatura abaixo, que permite devolver o número complexo, resultado da soma de dois complexos.

```
// @param n1, n2: Números complexos
complex somaCom(complex n1,complex n2)
```

3. (2.5V) No jogo do Bizz, os números naturais múltiplos de 7 chamam-se "Bizz", os acabados em 7 chamam-se "Buzz", e os que são "Bizz" e "Buzz" chamam-se "Bingo" (por exemplo o 7 e o 77). Elabore uma função que, dado um número natural N, devolva quantos "Bizz", "Buzz" e "Bingo" existem até N.

```
// @param N: Número inteiro
// @param *bizz, *buzz, *bingo: Número de naturais "bizz", "buzz" e "bingo", respetivamente
void jogoBizz(int N, int *bizz, int *buzz, int *bingo)
```

4. Considere o seguinte excerto:

```
#define MAX_ALUNOS 20
#define MAX_UCS 10

typedef struct uc {
  char nomeUC[15];
  float notaUC;
}UC;
```

```
typedef struct aluno {
  char nome[50];
  UC cadeiras[MAX_UCS];
  float mediaUCs;

}Aluno;
```

a) (2.0V) Elabore as instruções que lhe permite criar o vetor turma com um número de alunos definido pelo utilizador.

b) (2.0V) Elabore uma função que calcule, para cada aluno, o campo mediaUCs.

```
// @param turma: Vetor de alunos
// @param n: Número de alunos no vetor anterior
                void notaMediaAlunos(Aluno *turma, int n)
```

c) (2.5V) Implemente a função PosicaoDaMelhor que determina (e devolve) a maior média e respetiva posição/índice obtida por um aluno da turma.

```
// @param turma: Vetor de alunos
// @param n: Número de alunos no vetor anterior
// @param maior: maior média

          int posicaoDaMelhor(Aluno *turma, int n, float *maior)
```

d) (2.0V) Desenvolva uma função que escreva, num ficheiro de texto, o(s) nome(s) e nota(s) do(s) aluno(s) de uma determinada UC.

```
// @param turma: Vetor de alunos
// @param n: Número de alunos no vetor anterior
// @param nome: Nome da UC
// @param f: ficheiro a gravar

          void infoUC(Aluno *turma, int n, char *nome, FILE *f)
```

5. (2.0V) Considere a função mostra. Indique e justifique o que é mostrado no monitor caso a função seja invocada usando mostra(9,2).

```
void mostra(int n, int i){
 if ( (n>1) && i>0 ){
   mostra(n-i,i-1);
 }
 printf("\n%d",n+i);
}
```

6. (2.5V) A função PesqBin, abaixo apresentada, permite procurar, de forma eficiente, um elemento num vetor.

```
int PesqBin (int *vect, int indInf, int indSup, int chave)
{
   int indMeio;
   if (indInf > indSup)
      return -1;
   else
   {
      indMeio =(indInf + indSup) / 2;
      if (chave < vect[indMeio])
         return PesqBin(vect, indInf, indMeio-1, chave);
      else if (vect[indMeio] == chave)
         return indMeio;
      else
         return PesqBin(vect, indMeio+1, indSup, chave);
   }

}
```

Considere o vetor declarado e a instrução seguinte:

```
int vet[12]={1, 3, 4, 9, 10, 12, 14, 15, 16, 20, 25, 30};
printf("A posição do vetor é: %d\n", PesqBin(vet, 0, 9, 25));
```

O que vai ser escrito no monitor? Justifique a sua resposta, passo a passo.