

Programação Imperativa – CC (1º ano)

Teste final

25 de Junho de 2010 (9h00)

Dispõe de **2:00 horas** para realizar este teste.

Questão 1 (cálculo numérico) [4v = 2 + 2]

- a) Defina uma função `int Uns (int x)` que, dado um número inteiro calcula quantos dos bits da sua representação binária são 1. Exemplo:

```
...
x = Uns(7);
printf("%d",x);
...
```

Este excerto de código colocaria o valor 3 no monitor.

- b) Defina uma função `int mmc (int x, int y)` que, dados dois números inteiros positivos, calcula o seu mínimo múltiplo comum. Exemplo:

```
...
a = mmc(6,12);
printf("%d",a);
...
```

Este excerto de código colocaria o valor 24 no monitor.

Questão 2 (arrays) [4v = 2 + 2]

- a) Defina uma função que recebe como parâmetro, dois arrays unidimensionais (vectores), de comprimento máximo M , representando dois Polinómios $P1$ e $P2$ de grau máximo $M-1$ e dá como resultado Verdadeiro, se $P2$ for a derivada de $P1$, e Falso caso contrário;
- b) Defina uma função `int remCons (char *s)` que recebe uma string (terminada com o caracter `'\0'`) e apaga todos os caracteres iguais ao anterior. A função pretendida deverá fazer esta operação sem usar um array auxiliar e deve retornar o comprimento da string resultante.

Questão 3 (pergunta dada) [2v]

Considere uma árvore binária de procura definida pelo seguinte tipo abstracto de dados em C:

```
typedef struct sArvBin
{
    int valor;
    struct sArvBin *esq, *dir;
}
*ArvBin, NodoArvBin;
```

Especifique uma função que faz uma listagem *inorder* dos nodos da árvore (na especificação desta função está proibida a utilização de recursividade quer na função principal quer em qualquer função auxiliar).

Questão 4 (estruturas dinâmicas) [10v = 2 + 1 + 2 + 2,5 + 2,5]

Considere as seguintes estruturas de dados usadas para suportar um sistema de informação para registo da avaliação prática de grupos de alunos em disciplinas de programação:

```

#define MAXTRAB 7

typedef struct sAluno
{
    char *nome;
    char *numero;
    int nota[MAXTRAB];
} Aluno;

typedef struct sListAluno
{
    Aluno a;
    struct sListAluno *seg;
} *ListAluno, NodoListAluno;

typedef struct sGrupo
{
    char *idgrupo;
    ListAluno alunos;
} Grupo;

typedef struct sAvalPrat
{
    Grupo g;
    struct sAvalPrat *seg;
} *AvalPrat, NodoAvalPrat;

```

Defina então as seguintes funções em C:

- a) Defina uma função void `listar (AvalPrat av)` que recebe uma estrutura do tipo *AvalPrat* e a lista completamente no monitor (sugestão: escreva uma função de listagem para cada elemento estrutural);
- b) Defina uma função `Aluno consAluno (char *nome, char *num, int nota)` que recebe uma string com um nome, uma string com um numero, um inteiro e constrói um registo de aluno que é devolvido como resultado;
- c) Defina uma função `AvalPrat consAvalPrat (Grupo g, AvalPrat av)` que recebe um *Grupo*, uma estrutura *AvalPrat* e dá como resultado uma nova estrutura *AvalPrat* com o Grupo lá inserido;
- d) Defina uma função `float GetMediaAluno (AvalPrat av, char *nome)` que recebe uma estrutura *AvalPrat* e o nome de um aluno e dá como resultado um valor real correspondente à média dos trabalhos avaliados até esse momento (considere que o array de notas é inicializado a 0);
- e) Defina uma função `ListAluno GetAlunos (AvalPrat av)` que recebe uma estrutura *AvalPrat* e dá como resultado uma estrutura do tipo *ListAluno* em que **os alunos se encontram ordenados alfabeticamente por nome**;