Escola Superior de Tecnologia e Gestão
Politécnico de Coimbra

Ficha de Trabalho n.º 4

Estruturas de Dados | Programação II
LEI | LSTI
2022/2023

Ponteiros

1. Considere as seguintes três versões da função *troca* e as cinco possibilidades de chamada de cada uma dessas funções. Dependendo da forma como os dois elementos (função e chamada) são combinados, o resultado da sua execução poderá ser diferente.

```
int main()
{
      int a, b;
      unsigned long int i, j;

      i = &a;
      j = &b;
      // troca...
}
```

Funções

A

```
void troca(int *x, int *y)
{
      int aux;

      aux = *x;
      *x = *y;
      *y = aux;
}
```

B

```
void troca(int *x, int *y)
{
      int * aux;

      aux = x;
      x = y;
      y = aux;
}
```

C

```
void troca(int *x, int *y)
{
      int aux;

      aux = *x;
      x = *y;
      *y = aux;
}
```

Chamadas da função

I

```
int main()
{
      // ...
      troca(i,j);
}
```

II

```
int main()
{
      // ...
      troca(a,b);
}
```

III

```
int main()
{
      // ...
      troca(&i,&j);
}
```

IV

```
int main()
{
      // ...
      troca(*i,*j);
}
```

V

```
int main()
{
      // ...
      troca(*a,*b);
}
```

Considere as seguintes condições iniciais (antes de chamar a função *troca*) dos campos de memória do computador:

Nome de variável						*a*	*b*		*i*		*j*	
Endereço "físico"	…	5	…	12	…	1001	1002	…	2220	…	2224	…
Valor		1001		1002		5	12		1001		1002	

Indique quais os valores nos campos de memória 5, 12, 1001, 1002, 2220 e 2224 depois da execução de cada uma das 15 combinações possíveis do programa.

Por exemplo, combinando a função *A* com as instruções *I* obtém-se

Nome de variável						*a*	*b*		*i*		*j*	
Endereço "físico"	…	5	…	12	…	1001	1002	…	2220	…	2224	…
Valor		1001		1002		12	5		1001		1002	

Ponteiros, Tabelas e *Strings*

2. Usando ponteiros, elabore subprogramas que manipulem uma matriz de reais com dimensão 20×20:
   a. Inicializar a matriz com o –1;

```
void inicializaMatrizA(float *pm)
```

   b. Colocar numa posição da matriz (*i* , *j*) o valor indicado pelo utilizador;

```
void atribuiValorAij(float *pm, int i, int j)
```

3. Usando ponteiros, elabore novas versões para as funções
   a. *strlen* (para contar os caracteres de uma string)

      existente:

```
size_t strlen(const char *s)
```

      nova versão:

```
int tamanho_str(char *ps)
```

   b. *strncpy* (usada para copiar parte de uma string)

      existente copia n caracteres de uma string:

```
char * stpncpy(char * dst, const char * src, size_t len)
```

      nova versão deve copiar parte de uma string indicando a partir de onde pretende copiar e quantos caracteres:

```
char * copia_substr(char * src, int i, int len)
```

Ponteiros para tabelas de structs

4. Relembre o exercício 3. da Ficha de Trabalho n.º 3, no qual uma pequena empresa familiar pretendia organizar os dados dos seus funcionários (número, nome, tarefa, salário). Foi definida uma estrutura de dados para guardar os dados de cada funcionário e declarada uma tabela para armazenar os dados de todos os funcionários.

   Em vez de definir uma tabela, defina um ponteiro para *n* funcionários (*n* indicado pelo utilizador) e refaça os seguintes subprogramas:
   i. introduzir os dados de um funcionário na lista de funcionários;
   ii. listar todos os funcionários e respetivos dados;
   iii. listar os funcionários com salário superior a 500€;
   iv. procurar e apresentar todos os dados de um funcionário, usando o seu nome;
   v. atualizar os dados de um funcionário (usando o seu número).