

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

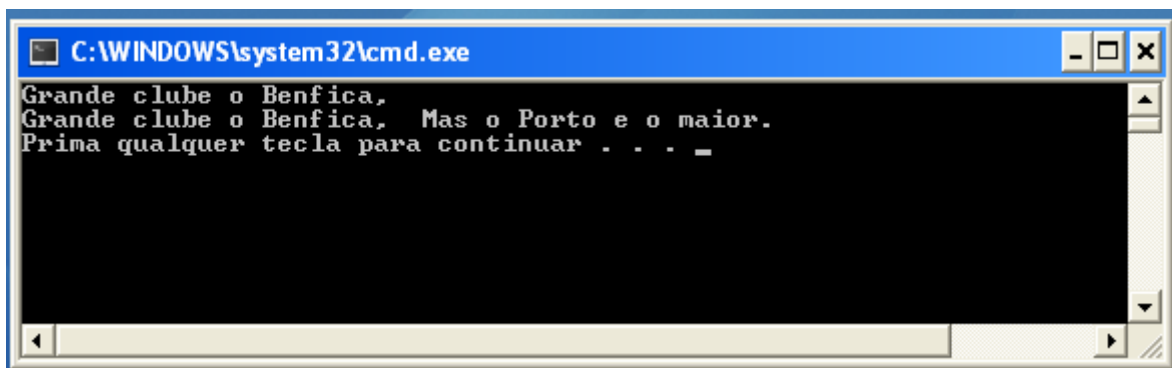
1.º Ano/2.º Semestre

Unidade Curricular: Programação Orientada a Objectos

Docente: Frederica Gonçalves

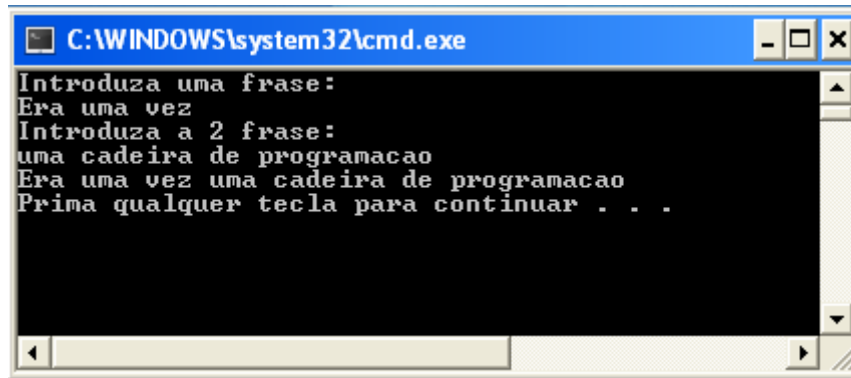
COLECTÂNEA DE EXERCÍCIOS - III

1. Faça um programa que utilize uma função para acrescentar uma string a outra.



```
#include <iostream>
using namespace std;
void main() {
    char frase1[100]="Grande clube o Benfica, ";
    char frase2[100]=" Mas o Porto e o maior." ;
    cout<< frase1 <<'\n';
    strcat (frase1, frase2);
    cout<< frase1 <<'\n';
}
```

2. Faça um programa que peça ao utilizador para introduzir duas frases e efectue a concatenação entre as duas strings.

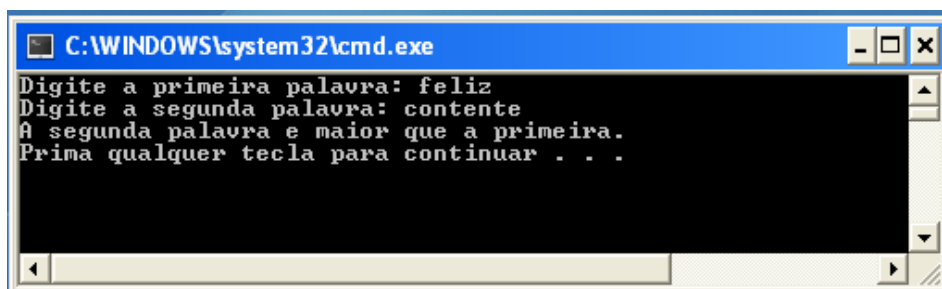


```
#include<iostream>
using namespace std;

void main(){
    char str1[100];
    char str2 [100];
    cout<< "Introduza uma frase: ";
    gets (str1);
    cout<<"Introduza a 2 frase: ";
    gets (str2);
    strcat (strcat(str1," "),str2 );
    cout<< str1<<endl;
}
```

.3. Faça um programa que dadas duas palavras introduzidas pelo utilizador, determine:

- a. Se as palavras são iguais;
- b. Caso as palavras sejam diferentes, qual delas tem maior comprimento.



```
#include <iostream>
using namespace std;

void main(){
    char palavra1[15], palavra2[15];

    cout<<"Digite a primeira palavra: " ;
    gets(palavra1);
    cout<<"Digite a segunda palavra: " ;
    gets(palavra2);

    if(strcmp(palavra1,palavra2)==0)
        cout << "Palavras sao Iguais.\n";
    else {
```

Cofinanciado por:

```

        if(strlen(palavra1)>strlen(palavra2))
            cout << "A 1a palavra e maior que a segunda.\n";
        else
            if(strlen(palavra1)<strlen(palavra2))
                cout << "A 2a palavra e maior que a primeira.\n";
            else
                cout << "As palavras tem o mesmo comprimento mas nao
                sao iguais.\n";
    }
}

```

4. Faça um programa que seja você próprio a definir o *array* que vai utilizar, com 7 valores inteiros, por exemplo **v[7] = {1,2,3,4,5,6,7}**. Usando a técnica da aritmética de ponteiros, com o operador de indirecção (*), obtenha o output da seguinte forma:

- Apresente o valor do índice [0], apontando para o índice [1];
- Apresente o valor do índice [3], mantendo o apontador para o índice [1];
- Apresente o valor do índice [2], antecipando o apontador para o índice [2];
- Apresente o valor do índice [6], apontando para o índice [3];
- Apresente o valor do índice [0], apontando para o índice [4];
- Apresente todos os valores do *array*, separados por vírgulas, e após o último coloque um ponto final.

Ex: Output do array v[7]: 1 4 3 7 1 1,2,3,4,5,6,7.

```

#include <iostream>
#include <ctime>

using namespace std;

main(){
    int v[7] = {1,2,3,4,5,6,7};
    int *p = v;
    int i;
    cout <<*p++ <<"\n";
    cout << *(p+2) <<"\n";
    cout << *++p <<"\n";
    cout << *(p+4) <<"\n";
    *p++;
    cout << *(p-3) <<"\n";
    *p++;
    for(i = -4; i < 2; i++)
        cout << *(p+i) << ", ";
    cout << *(p+i) << ".\n";
    system("pause");
}

```

Cofinanciado por:

