

Licenciatura em Informática

Exame Tipo

P I

Solução

Atualizado em 25-01-2018

Data ??

Às ??:00H

Duração:

1:30H

Nota:

As respostas podem ser feitas tanto no próprio enunciado como na Folha de Teste (à sua escolha).

Cotação:

Perguntas 1, 2, 3, 4, 5 e 6: **1** Valor cada (6 Valores).

Restantes perguntas: **1, 75** Valores cada (14 Valores).

1. Dado o seguinte excerto de código, calcule os resultados pretendidos, após a execução da última instrução, conforme abaixo indicado.

```
int i=2, k=2, j=2;
```

```
k = i++ * ++k + j--;
```

Resposta:

i ---- **3**

j ---- **1**

k ---- **8**

2. Qual é o resultado obtido após a resolução da seguinte expressão:

```
! (( 6 && !(2 && 3))) == 0
```

Resposta:

0 (Falso)

3. Considere o seguinte código:

```
main()
{
    int ch;

    ch = 'b' + 1;

    printf("%c\n", ch);
}
```

Qual é o output apresentado no ecrã?

- ☒ c
- ☐ Indeterminado
- ☐ Erro de compilador
- ☐ Em função da implementação do compilador

4. Considere o seguinte trecho de código:

```
int main() {
    int i1, i2, *p1, *p2;

    i1 = 3;

    p1 = &i1;

    i2 = *p1 / 2 + 10;

    p2 = p1;

    printf("i1 = %d, i2 = %d, *p1 = %d, *p2 = %d\n", i1, i2, *p1, *p2);
}
```

Escreva o output apresentado no ecrã (Resposta):

i1 = 3

i2 = 11

***p1 = 3**

***p2 = 3**

5. Considere os seguintes comandos. Escolha o que considera correto para declarar uma variável designada por aluno, definida a partir da estrutura registo:

- ☐ type aluno : registo;
- ☐ struct aluno;
- ☒ struct registo aluno;
- ☐ declare aluno as type registo;

6. Qual dos seguintes comandos retorna o conteúdo da localização de memória apontada pelo pointer p?

- ☐ p;
- ☒ *p;
- ☐ &p;
- ☐ **p;

7. Dado o seguinte vetor de 5 doubles, escreva um **excerto** de código a fim de **percorrerlo** e mostrar no ecrã os **endereços** de Memória de cada elemento

```
double nums[] = {5.2, 3.5, 8.3, 2.9, 1.2};
```

Resposta:

```
double nums[] = {5.2, 3.5, 8.3, 2.9, 1.2};
```

```
int i = 0;
```

```
while (i < 5)
```

```
{  
    printf ("\n%ld", &nums[i]);
```

```
    i++;
```

```
}
```

Ou então, se preferisse usar uma outra técnica (não obrigatória):

```
double nums[] = {5.2, 3.5, 8.3, 2.9, 1.2};
```

```

int i = 0;

double *p;

p = &nums[0];

while ( i < 5)
{
    printf ("\n%d", p);

    i++;
    p++;
}

```

8. Dado o seguinte excerto de código, esclareça se funciona. Em caso afirmativo, diga qual é o resultado produzido no ecrã. Em caso negativo, indique o(s) erro (s).

```

int m[3][3]= {
    {2, 2, 1},
    {2, 2, 3},
    {0, 2, 3}
};
int i = 0;
int k = 0;
int j = 2;

while (i < 3)    {

    k = k + m[ i ][ i ];
    k = k + m[ i ][ j ];
    i++;
    j--;

}    cout << k << endl;

```

Resposta:

K = 10

9. Escreva uma função que receba como argumento um vetor de cinco inteiros e que devolva o valor do último elemento (observe que **somente** a função é pedida para criar).

Resposta (a resposta é somente esta e nada mais porque não é pedido) :

```

int f1( int vals[] )
{
    return vals[ 4 ];
}

```

10. Dado a seguinte função, onde p1 é um ponteiro que aponta para o primeiro elemento de um determinado vetor de inteiros e p2 aponta para o último elemento do mesmo vetor, escreva o código necessário para a função devolver o número de bytes contidos no vetor.

```
int contaBytes(int *p1, int *p2)
{

}
```

Resposta:

```
int contaBytes( int *p1, int *p2 )
{
    int k = 0;

    while(p1++ <= p2) k++; // conta quantos elementos o array contém

    return (k * sizeof(int)); // Número de elementos * tamanho do inteiro
}
```

11. Escreva uma função que receba uma string e que devolva o endereço da primeira letra *a*. Caso a string não contenha letras *a*, então a função deve devolver um ponteiro NULL.

Resposta:

```
char * f1( char m[] )
{
    int i = 0;
    while ( m[ i ] != '\0' )
    {
        if ( m[ i ] == 'a' ) return &m[ i ];
        i++;
    }

    return NULL;
}
```

12. Escreva uma classe designada por cliente com os seguintes members:

idade (**private**)

nome (**public**)

A classe deve ter dois construtores, um construtor para inicializar somente o member idade e outro para inicializar os members idade e nome.

Resposta:

```
class cliente
{
    private:
        int idade;

    public:
        char nome[ 30 ];

        cliente ( int i, char *m )
        {
            idade = i;

            strcpy(nome, m);
        }

        cliente ( int i )
        {
            idade = i;
        }

        cliente () // construtor default
        {
        }

} // Fecha a classe cliente
```

13. Reescreva a classe cliente utilizando um método apropriado de forma a permitir obter o valor do data member private idade.

Resposta:

```
class cliente // esta parte da classe mantém-se, mas
{
    private:
        int idade;

    public:
        char nome[ 30 ];

        cliente ( int i, char *m )
        {
            idade = i;

            strcpy(nome, m);
```

```

    }

    cliente ( int i )
    {
        idade = i;
    }

    cliente () // construtor default
    {

    }

    int getIdade() // mas acrescenta-se assim este método à classe cliente
    {
        return idade;
    }

} // fecha a classe cliente

```

14. Em referência à classe criada na pergunta anterior, crie na função main dois objetos. Um objeto deve inicializar apropriadamente os data members idade e nome com o valor 20 e ISTECC, respetivamente, mas através dos construtores já implementados.

Outro objeto deve ler e imprimir adequadamente o valor do campo idade da classe, sem uso de construtores.

Resposta:

```

void main()
{
    cliente cli1(20, "ISTEC");

    cliente cli2;

    printf("\n Idade do cli2 = %d", cli2.getIdade());

}

```

Nota:

Outra pergunta tipo que pode sair em vez de uma das perguntas acima:

O seguinte excerto de programa contém erros.

Identifique-os (**não é necessário refazer o código**, somente é pedido para identificá-los).

```
#include <iostream >
```

```
#define MACRO QUADRADO(x) (x * x); // A palavra MACRO está a nais
```

```
struct cliente  
{  
    long num;  
    char nome[30];  
};
```

```
struct cliente cli;  
struct cli *p;
```

O código abaixo está fora de contexto (não pertence a nenhuma função!

```
cout << endl << "Insira numero do Cliente: ";  
fflush(stdin);  
cin >> p.num;
```

```
cout << endl << "Insira Nome do Cliente: ";  
fflush(stdin);  
gets(p.nome);
```

```
cout << endl << "Dados Introduzidos: " << endl << endl;
```

```
cout << "Num do Cliente: " << p.num << endl;  
cout << "Nome do Cliente: " << p.nome << endl;
```

```
int void main()  
{  
    struct cliente cli;  
    &cli;  
  
    float lado;  
    cout << "\nDigite o comprimento do lado do quadrado: ";  
  
    cin << lado;  
  
    cout << "O Quadrado com o lado " << lado << " = " << quadrado(lado) << endl;
```

Código abaixo é absurdo: Trata-se de uma função com o mesmo nome de uma macro!!!!

E sendo uma função não pode estar inserida dentro de outra função seja qual for.

```
int QUADRADO (double float x)  
{  
    return x * x;  
}  
system("pause");  
  
} // Fecha main
```