

1)

```
main(){
    int myCount = 0;
    while (myCount < 10)
    {
        printf("%d",myCount+1);
    }
}
```

Assinale a alternativa que apresenta, corretamente, o que esse trecho de código fará ao ser executado.

- a) Mostrará na tela os valores de 0 a 9.
- b) Mostrará na tela os valores de 1 a 10.
- c) Escreverá na tela myCount por 10 vezes.
- d) Escreverá na tela 0 por 10 vezes.
- e) Entrará em looping infinito.

2)

Trecho 1

```
main()
{
    int mat[2][2] = {{1,2},{3,4}};
    int i,j;

    for (i=0;i<2;i++)
        for (j=0;j<2;j++)
            printf("%d\n",mat[i][j]);

    system("pause");
}
```

Trecho 2

```
main()
{
    int mat[2][2] = {{1,2},{3,4}};
    int *p = &mat[0][0];
    int i;

    for (i=0;i<4;i++)
        printf("%d\n",*(p+i));

    system("pause");
}
```

Com base nos trechos de código acima, fale qual será o resultado e explique.

3)

Seja o seguinte trecho de programa:

```
main(){
    int i=3,j=5;
    int *p, *q;
    p = &i;
    q = &j;
    ...
}
```

Qual é o valor das seguintes expressões ?

- a)  $p = \&i$
- b)  $*p - *q$
- c)  $**\&p$
- d)  $3* - *p/(*q)+7$

4)

Qual será a saída deste programa supondo que "i" ocupa o endereço 1000 na memória?

```
main() {
    int i = 5, *p;
    p = &i;
    printf("%p %d %d %d %d\n", p, *p+2, **&p, 3**p, **&p+4);
}
```

5) Assumindo que pulo[] é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento da vetor?

- a)  $*(pulo + 2)$
- b)  $*(pulo + 4)$
- c)  $pulo + 4$
- d)  $pulo + 2$

6) Suponha as declarações: int mat[4], \*p, x; Quais expressões são válidas? Justifique:

- a)  $p = mat + 1;$
- b)  $p = mat++;$
- c)  $p = ++mat;$
- d)  $x = (*mat)++;$

7) Faça um programa que lê n números decimais (n fornecido pelo usuário), armazenando-os em um vetor. Logo após, uma função deve retirar os números duplicados que eventualmente possam existir neste vetor, deixando apenas uma ocorrência de cada número. Ao final, o vetor resultante deve ser impresso na tela.

8) Faça um programa que leia uma string (de 80 caracteres) chamada linha e, com uma função, identifique cada palavra (substring) desta linha copiando-a para um novo vetor. Ao final, o programa deve imprimir as palavras separadas, uma palavra por linha.

Dicas: i) faça uma função para identificar o fim e/ou o início de cada substring em linha

9) Considere a função desenvolvida na Linguagem C, a seguir:

```
char *Teste (char *s1, const char *s2) {  
    char *aux=s1;  
    while (*s1) s1++;  
    for ( ; (*s1 = *s2) != '\0' ; s1++, s2++);  
    return aux;  
}
```

O seu objetivo é:

- a) Copiar o conteúdo da região de memória referenciada pelo identificador s1 para a região de memória referenciada pelo identificador s2.
- b) Atribuir o valor '\0' para todas as posições de memória entre o endereço referenciado pelo identificador s1 até a região de memória referenciada pelo identificador s2.
- c) Comparar o conteúdo de memória que se inicia na posição referenciada pelo identificador s1 e ir até a ocorrência de um valor '\0' com o conteúdo da região de memória referenciada pelo identificador s2.
- d) Substituir os elementos armazenados na região de memória referenciada pelo identificador s1 pelos elementos armazenados na região de memória referenciada pelo identificador s2.
- e) Copiar os elementos contidos na região de memória referenciada pelo identificador s2 após os elementos armazenados na região de memória referenciada pelo identificador s1.

10) Considere a função desenvolvida na Linguagem C.

```
int F1 ( unsigned int n)  
{  
    if(n==0) return n;  
  
    int i , j;  
    for ( i = j = 1; i<2 *n-1 ; i+=2, j +=i);  
    return j;  
}
```

- a) Explique esta função;
- b) Calcule o valor de j para o valor de entrada n = 10;
- c) Implemente uma função similar utilizando 2 loops for;