

1. Quando não se escreve o protótipo de uma função... (marque a alternativa correta):
 - (a) A chamada da função poderá ser feita em qualquer hipótese
 - (b) A chamada da função não poderá ser feita em qualquer hipótese.
 - (c) O programa funcionará de qualquer jeito.
 - (d) É preciso definir a função antes do programa principal.
 - (e) N.A
2. Em relação ao código a seguir, marque a alternativa que representa o resultado final impresso pelo programa:

```
#include <iostream>

using namespace std;

int i, *iptr;

int main()
{
    i=10;
    iptr=&i;
    (*iptr)++;
    i=i+2;
    cout << i + (*iptr) + 2<< endl;
    return 0;
}
```

- (a) 25
 - (b) 20
 - (c) 28
 - (d) 26
 - (e) N.A
3. Uma estrutura de dados pode ser Estática ou Dinâmica. No primeiro caso os limites são:
 - (a) Determinados pelo usuário
 - (b) Determinados pelos ponteiros
 - (c) Determinados pelo problema em que serão utilizados
 - (d) Indeterminados
 - (e) Determinados pela memória
4. Marque a alternativa que representa o valor final da variável i:

```
#include <iostream>

using namespace std;

int i, *iptr;

int main()
{
    i=10;
```

```
    iptr=&i;  
    i=(*iptr)++;  
    cout << i << endl;  
    return 0;  
}
```

- (a) 10
- (b) 11
- (c) 12
- (d) 9
- (e) N.A

5. A partir da estrutura heterogênea a seguir:

```
struct aluno {  
    string nome;  
    float nota;  
};
```

Suponha ainda que exista um vetor desta estrutura, definida como:

```
aluno vet[50];
```

Marque a alternativa em que é atribuída a nota 8.8 para o quinto elemento deste vetor:

- (a) aluno[4].nota = 8.8
- (b) aluno[5]->nota = 8.8
- (c) struct[4].aluno.nota = 8.8
- (d) vet[5]->nota = 8.8
- (e) vet[4].nota = 8.8

6. Marque a alternativa que representa a melhor forma de fazer a leitura de 13 valores consecutivos para a estrutura heterogênea a seguir:

```
struct aluno {  
    string nome;  
    float nota;  
};
```

Sabendo que a declaração é feita por:

```
aluno alunos[13];
```

- (a) for (int i=0; i<13; i++) {cin>>aluno[i].nome; cin>>aluno[i].nota;}
- (b) for (int i=0; i<=13; i++) {cin>>aluno[i]->nome; cin>>aluno[i]->nota;}
- (c) for (int i=0; i<13; i++) {cin>>alunos[i]->nome; cin>>alunos[i]->nota;}
- (d) for (int i=0; i<13; i++) {cin>>alunos[i].nome; cin>>alunos[i].nota; }

(e) N.A

7. Escreva uma função que receba um número inteiro e imprima se o número é par ou impar. Use o protótipo: **parImpar(int)**;
8. Escreva uma função que receba as notas A1 e A2, ambas inteiras e positivas. A função deve imprimir "Aluno aprovado" caso a média seja maior ou igual a 7.0 ou imprimir "Aluno apto para a terceira prova" caso a média esteja entre 7.0 e 5.0, inclusive, caso contrário, imprimir "Fazer novamente a disciplina".
9. Escreva o código de um programa que leia 10 posições inteiras e imprima os valores na tela. Usar modularização no programa.