Algoritmos e Programação II

IV Lista de Exercícios: Ponteiros

Prof. Evandro C. R. Rosa
UNIVALI

Nome Completo: _____ Código de Aluno: _____

1. Responda sucintamente:
   (a) O que faz o operador de desreferência (`*`)?
   (b) Veja o código a seguir:

```
int x = 7;
int *iptr = &x;
```

   i. O que será exibido se você enviar a expressão `*iptr` para `cout`?
   ii. E se você enviar a expressão `iptr` para `cout`?

   (c) Quais operações matemáticas são permitidas com ponteiros?
   (d) Qual é a finalidade do operador `new`?
   (e) Qual é a finalidade do operador `delete`?
   (f) Em quais circunstâncias é possível retornar um ponteiro de uma função?

2. Os códigos abaixo contêm um ou mais erros. Indique quais são:
   (a)

```
int ptr = nullptr;
```

   (b)

```
int x, *ptr = nullptr;
&x = ptr;
```

   (c)

```
int x, *ptr = nullptr;
*ptr = &x;
```

   (d)

```
int x, *ptr = nullptr;
ptr = &x;
ptr = 100; // Armazena 100 em x
cout << x << endl;
```

   (e)

```
int numeros[] = {10, 20, 30, 40, 50};
cout << "O terceiro elemento do array é ";
cout << *numeros + 3 << endl;
```

   (f)

```
int valores[20], *ponteiro_int = nullptr;
ponteiro_int = valores;
ponteiro_int *= 2;
```

   (g)

```
float nivel;
int ponteiro_float = &nivel;
```

(h)
```
int *ponteiro_int = &numero;
int numero;
```

(i)
```
void dobrar_valor(int valor)  {
    *valor *= 2;
}
```

(j)
```
int *ponteiro_int = nullptr;
ponteiro_int = new int[100]; // Aloca memória
delete ponteiro_int; // Libera memória
```

(k)
```
int *obter_numero()  {
    int numero;
    cout << "Digite um número: ";
    cin >> numero;
    return &numero;
}
```

(l)
```
struct TresValores {
    int a, b, c;
};
int main () {
    TresValores s, *s_ponteiro = nullptr;
    s_ponteiro = &s;
    *s_ponteiro.a = 1;
    return 0;
}
```

3. Em estatística, a moda de um conjunto de valores é o valor que ocorre com maior frequência. Escreva uma função que aceite os seguintes argumentos:

   - Um array de inteiros
   - Um inteiro que indica o número de elementos no array

   A função deve determinar a moda do array, ou seja, qual valor no array ocorre com mais frequência. A moda é o valor que a função deve retornar. Se o array não tiver moda (nenhum valor ocorre mais de uma vez), a função deve retornar -1 – Assuma que o array sempre conterá valores positivos. Demonstre a função em um programa completo.

4. Escreva uma função que aceite um array de inteiros e o tamanho do array como argumentos. A função deve criar uma cópia do array, exceto que os valores dos elementos devem ser invertidos na cópia. A função deve retornar um ponteiro para o novo array. Demonstre a função em um programa completo.

5. Escreva uma função que aceite um array de inteiros e o tamanho do array como argumentos. A função deve criar um novo array que seja o dobro do tamanho do array argumento. A função deve copiar o conteúdo do array argumento para o novo array e inicializar os elementos não utilizados do segundo array com 0. A função deve retornar um ponteiro para o novo array. Demonstre a função em um programa completo.

6. Escreva um programa que aloca dinamicamente um array grande o suficiente para armazenar uma quantidade de notas fornecida pelo usuário. Após a inserção de todas as notas (de 0 a 10), o array deve ser passado para uma função que as ordena em ordem crescente. Outra função deve ser chamada para calcular a média das notas. O programa deve exibir a lista ordenada de notas e a média com títulos apropriados.

7. Modifique o problema acima para que a menor nota seja descartada. Essa nota não deve ser incluída no cálculo da média.

8. Modifique o programa do problema acima para permitir que o usuário insira pares nome-nota. Para cada aluno que fizer uma prova, o usuário deve digitar o nome do aluno seguido pela nota da prova do aluno. Modifique a função de ordenação para que ela aceite um array contendo os nomes dos alunos e um array contendo as notas dos testes dos alunos. Quando a lista ordenada de notas for exibida, o nome de cada aluno deve ser exibido junto com sua nota.

9. Escreva um programa que use uma estrutura para armazenar os seguintes dados:

   - nome: Nome do aluno
   - notas: Ponteiro para um array de notas
   - media: Média das notas

   O programa deve manter uma lista de notas para um grupo de alunos. Deve perguntar ao usuário quantas notas há e quantos alunos existem. Em seguida, deve alocar dinamicamente um array de estruturas. O membro `notas` de cada estrutura deve apontar para um array alocado dinamicamente que conterá as notas. Após a alocação dinâmica dos arrays, o programa deve solicitar o nome e todas as notas para cada aluno. A média das notas deve ser calculada e armazenada no membro `media` de cada estrutura. Depois que todos esses dados forem calculados, uma tabela deve ser exibida na tela listando o nome de cada aluno e a média das notas.

   Validação de Entrada: Não aceite números negativos para nenhuma nota.