

JEAN ALVES ROCHA

RA: 2313057

Lista de Exercícios sobre Alocação Dinâmica

Para os exercícios abaixo, substitua as variáveis do tipo int, double, char e float por alocação dinâmica. Se houver ponteiros no código, mantenha-os.

1. Algoritmo simples 1:

```
int main(int argc, char** argv) {  
    int a;  
    a=3;  
    printf("Valor: %d",a);  
    return 0;  
}
```

```
int main(int argc, char** argv) {  
    int *b;  
    b = (int*)malloc(sizeof(int));  
    *b = 3;  
    printf("Valor: %d",*b);  
    free(b);  
    return 0;  
}
```

2. Algoritmo simples 2:

```
int main(int argc, char** argv) {  
    double x;  
    x=3.2;  
    printf("Valor: %lf",x);  
    return 0;  
}
```

```
int main(int argc, char** argv) {  
    double *z;  
    z = (double*) malloc (sizeof(double));  
    *z = 3.2;  
    printf("Valor: %lf",*z);  
    free(z);  
    return 0;  
}
```

### 3. Algoritmo simples 3:

```
int main(int argc, char** argv) {  
    char z;  
    z='c';  
    printf("Valor: %c",z);  
    return 0;  
}
```

```
int main(int argc, char** argv) {  
    char *z;  
    z = (char*) malloc(sizeof(char));  
    *z='c';  
    printf("Valor: %c",*z);  
    free(z);  
  
    return 0;  
}
```

### 4. Algoritmo simples 4:

```
int main(int argc, char** argv) {  
    int a;  
    printf("\nDigite um valor:");  
    scanf("%d",&a);  
    printf("Valor: %d",a);  
    return 0;  
}
```

```
int main(int argc, char** argv) {  
    int *a;  
    a = (int*) malloc(sizeof(int));  
    printf("\nDigite um valor:");  
    scanf("%d",a);  
    printf("Valor: %d",*a);  
    free(a);  
    return 0;  
}
```

### 5. Algoritmo com função 1:

```
void apresentar(int b){  
    printf("Valor: %d",b);  
}
```

```

}

int main(int argc, char** argv) {
    int a;
    a=4;
    apresentar(a);
    return 0;
}

void apresentar(int *b){
    printf("Valor: %d",*b);
}

int main(int argc, char** argv) {
    int *a;
    a = (int*)malloc(sizeof(int));
    *a=4;
    apresentar(a);
    free(a);
    return 0;
}

```

#### 6. Algoritmo com função 2:

```

int ler(){
    int a;
    printf("Digite um valor");
    scanf("%d",&a);
    return a;
}

int main(int argc, char** argv) {
    int a;
    a=ler();
    printf("Valor: %d",a);
    return 0;
}

void ler(int *ponteiro2){
    printf("Digite um valor");
    scanf("%d",ponteiro2);
}

```

```

int main(int argc, char** argv) {
    int *ponteiro1;
    ponteiro1 = (int*)malloc(sizeof(int));
    ler(ponteiro1);
    printf("Valor: %d", *ponteiro);
    free(ponteiro1);
    return 0;
}

```

#### 7. Algoritmo com função 3:

```

int calcular(int a){
    a=a*2;
    return a;
}

int main(int argc, char** argv) {
    int a;
    a=3;
    a=calcular(a);
    printf("Valor: %d",a);
    return 0;
}

void calcular(int *a){
    *a=*a * 2;
}

```

```

int main(int argc, char** argv) {
    int *a;
    a = (int*)malloc(sizeof(int));
    *a=3;
    calcular(a);
    printf("Valor: %d",*a);
    free(a);
    return 0;
}

```

#### 8. Algoritmo com função 4:

```

int calcular(int a){
    a=a*2;
    return a;
}

```

```
int main(int argc, char** argv) {
    int a, dobro;
    a=3;
    dobro=calcular(a);
    printf("Valor: %d %d",a, dobro);
    return 0;
}
```

```
void calcular(int *a, int *dobro){
    *dobro =*a * 2;
}
```

```
int main(int argc, char** argv) {
    int *a, *dobro;
    a = (int*)malloc(sizeof(int));
    dobro = (int*)malloc(sizeof(int));
    a*=3;
    calcular(a, dobro);
    printf("Valor: %d %d",* a, *dobro);
    free(a);
    free(dobro);
    return 0;
}
```

---

#### 9. Algoritmo com Função 5:

```
int ler(){
    int a;
    printf("\nDigite um valor:");
    scanf("%d",&a);
    return a;
}
```

```
int calcular(int a){
    int i;
    i=2020 - a;
    return i;
}
```

```
void apresentar(int i){
    printf("\nIdade: %d",i);
}
```

```
int main(int argc, char** argv) {
```

```

        int ano, idade;
        ano=ler();
        idade=calcular(ano);
        apresentar(idade);
        return 0;
    }

    void ler(int *ano){
        printf("\nDigite um valor:");
        scanf("%d",ano);
    }

    void calcular(int *ano, int *idade){
        *idade=2020 - *ano;
    }

    void apresentar(int *i){
        printf("\nIdade: %d",*i);
    }

    int main(int argc, char** argv) {
        int *ano, *idade;
        ano = (int*)malloc(sizeof(int));
        ler(ano);
        idade = (int*)malloc(sizeof(int));
        calcular(ano, idade);
        apresentar(idade);
        free(ano);
        free(idade);
        return 0;
    }

```