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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

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

int \*a = (int \*)malloc(sizeof(int));

\*a = 3;

printf("Valor: %d", \*a);

free(a);

return 0;

}

2. Algoritmo simples 2:

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

double x;

x=3.2;

printf("Valor: %lf",x);

return 0;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

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

double \*x = (double \*)malloc(sizeof(double));

\*x = 3.2;

printf("Valor: %lf", \*x);

free(x);

return 0;

}

3. Algoritmo simples 3:

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

char z;

z=’c’;

printf("Valor: %c",z);

return 0;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

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

char \*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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

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

int \*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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

void apresentar(int \*b) {

printf("Valor: %d", \*b);

free(b); // Libere a memória alocada dinamicamente

}

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

int \*a = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

\*a = 4;

apresentar(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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

int \*ler() {

int \*a = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

printf("Digite um valor: ");

scanf("%d", a);

return a;

}

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

int \*a = ler();

printf("Valor: %d", \*a);

free(a); // Libere a memória alocada dinamicamente

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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

int calcular(int a) {

a = a \* 2;

return a;

}

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

int \*a = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

\*a = 3;

\*a = calcular(\*a);

printf("Valor: %d", \*a);

free(a); // Libere a memória alocada dinamicamente

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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

int calcular(int a) {

a = a \* 2;

return a;

}

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

int \*a = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

int \*dobro = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int para o resultado

\*a = 3;

\*dobro = calcular(\*a);

printf("Valor: %d %d", \*a, \*dobro);

free(a); // Libere a memória alocada dinamicamente

free(dobro); // Libere a memória alocada dinamicamente

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;

}

**Deve virar:**

#include <stdio.h>

#include <stdlib.h>

int ler() {

int \*a = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

printf("\nDigite um valor: ");

scanf("%d", a);

return \*a;

}

int calcular(int a) {

int \*i = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int para o resultado

\*i = 2020 - a;

return \*i;

}

void apresentar(int i) {

printf("\nIdade: %d", i);

}

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

int \*ano = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int

int \*idade = (int\*)malloc(sizeof(int)); // Aloca dinamicamente um int para o resultado

\*ano = ler();

\*idade = calcular(\*ano);

apresentar(\*idade);

free(ano); // Libere a memória alocada dinamicamente

free(idade); // Libere a memória alocada dinamicamente

return 0;

}