Lista de exercícios de Subprogramação

Aluno: Bruno Peternella da Silva ra: 119100

1. Modifique o programa abaixo e corrija os erros existentes na utilização de subprogramas que utilizam passagem de parâmetros por valor e por referência. Informe o resultado final impresso na tela.

#include <stdio.h>

float divisao(int x);

void multiplicacao(int x,int \*mult);

int main(){

int num = 5, mult;

float div;

multiplicacao(num,mult);

divisao(num);

printf("\n%d \* 2 = %d",num, mult);

printf("\n%d/2 = %3.2f",num,div);

}

float divisao(int x){

return (float) x/2;

}

void multiplicacao(int x,int mult){

\*mult = x \* 2;

}

1. A partir do programa abaixo e as informações de endereçamento de memória de variáveis, responda as seguintes questões:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variável | x | y | soma | r | v1 | v2 | res\_soma |
| Endereço | FFC6 | FAB5 | FZA4 | FYB3 | FWC3 | FFX6 | FGX9 |

1. #include <stdio.h>
2. void soma(int \*v1,int \*v2, int \*r);
3. int main(){
4. int soma = 0, x = 1, y = 4;
5. somar(&x, &y, &soma);
6. printf("soma: %i", soma);
7. }
8. int somar(int \*v1,int \*v2,int \*r){
9. int res\_soma = 0;

10. res\_soma = \*v1 + \*v2;

11. \*r = res\_soma;

12. }

1. Depois da linha 8 quais os endereços de:

|  |  |  |  |
| --- | --- | --- | --- |
| **&soma: FZA4** | **v1: FFC6** | **v2: FAB5** | **r: FZA4** |

1. Depois da linha 10 quais os endereços e valores de:

|  |  |  |
| --- | --- | --- |
| **soma: 0** | **res\_soma: 5** | **&res\_soma: FGX9** |

1. Depois da linha 11 quais os endereços e valores de:

|  |  |  |  |
| --- | --- | --- | --- |
| **&soma: FZA4** | **soma: 5** | **r: FYB3** | **\*r: FYB3** |

1. Qual o resultado final impresso em tela? 5
2. Dado o programa abaixo, complete as respostas:
3. #include <stdlib.h>
4. #include <stdio.h>
5. int validar\_entrada(int n);
6. void multiplicar(int y,float \*resultado);
7. float dividir(int n);
8. int x = 10, y = 3;
9. int main(){
10. float resultado;
11. int n = 0,x = 4, y = 5;
12. printf("\nDigite um numero:");
13. scanf("%d",&n);
14. if(eh\_par(n) == 1){
15. multiplicar(y,&resultado);
16. }else{

18. resultado = dividir(n);

19. }

20. printf("\nResultado %.2f", resultado);

21. }

22. int eh\_par(int n){

23. if((n % 2) == 0){

24. printf("\nNumero Par");

25. return 1;

26. }else{

27. printf("\nNumero Impar");

28. return 0;

29. }

30. }

31.

32. void multiplicar(int y,float \*resultado){

33. \*resultado = x \* y;

34. }

35.

36. float dividir(int n){

37. return (float) n / y;

38. }

1. As variáveis **x** e **y** da linha 8 estão no escopo **GLOBAL** da aplicação**.**
2. O parâmetro **n** do subprograma **eh\_par** é passado por **VALOR.**
3. O parâmetro **y** da função **multiplicar** é passado por **VALOR**  enquanto o parâmetro **resultado** é passado por **REFERÊNCIA**.
4. A variável **x** na função **multiplicar** é uma variável do escopo **GLOBAL**  enquanto **y** é uma variável do escopo **LOCAL.**
5. A variável **y** na função **dividir** pertence ao escopo **GLOBAL**.
6. Qual o valor da variável **resultado** supondo que temos n = 9 ? **3**.
7. A declaração da variável **x** e **y** na **linha 12** sobrepõem os valores de **x** e **y** da **linha 8** ? **NÃO** (sim/não).
8. O parâmetro **y** do método **multiplicar** e a variável **y** da **linha 8** são diferentes? Justifique **São! Pois como parâmetro, ele recebe o valor ou variável passada como parâmetro no momento em que a função é chamada.**
9. Escreva um algoritmo que receba uma palavra como entrada e em seguida chame um procedimento que na **mesma variável** de alocação, faça a inversão da palavra recebida. Utilize **passagem de parâmetros por referência** e um **vetor auxiliar** dentro do procedimento.