Linguagem de Programação II

Gabarito - Prova Escrita 2 - 2015.1

Questão 1

- a) Saída: c=1000, a=1000. A função 1 calcula a potência a^b e retorna o resultado. A função 2 calcula a potência a^b e armazena o resultado em a.
- b) func1 passagem por valor func2 passagem por referencia
- c) Passagem por valor: uma nova cópia da variável é criada na função, o valor da variável é enviado à função, o parâmetro original não pode ser modificado no corpo da função.

Passagem por referência: o endereço do parâmetro (referência) é enviado a função, o endereço pode ser utilizado para manipular a variável, o parâmetro original pode ser modificado no corpo da função.

Correção: Valor da questão 2.0, (a) 50%, (b) 20% e (c) 35%

Questão 2

```
#include <stdio.h>
  #include <math.h>
2
3
4
  typedef struct{
5
     float x;
    float y;
6
7
  }TPonto;
8
9
  typedef struct{
10
    TPonto centro;
11
     float r;
12
  }TCirc;
13
14
  void LePonto(TPonto *);
  void LeCirc(TCirc *);
16 int VerificaPontoInterno(const TPonto *, const TCirc *);
17
18
  int main(){
19
    TPonto P;
20
     TCirc C;
21
     int res;
22
23
    LePonto(&P);
24
     LeCirc(&C);
25
     res = VerificaPontoInterno(&P, &C);
26
     if (res)
       printf("Ponto Interno.\n");
27
28
     else
```

```
29
       printf("Ponto Externo.\n");
30
    return 0;
31
  }
32
  void LePonto(TPonto *AuxP){
33
34
     printf("Digite a ordenada do ponto: ");
35
    scanf("\%f", &(AuxP->x));
36
    printf("Digite a abcisa do ponto: ");
37
    scanf("%f", &(AuxP->y));
38
39
  void LeCirc(TCirc *AuxC){
     printf("Digite a ordenada do centro: ");
40
     scanf("\%f", &(AuxC->centro.x));
41
42
     printf("Digite a abcisa do centro: ");
43
    scanf("%f", &(AuxC->centro.y));
     printf("Digite o raio:");
44
45
     scanf("\%f", &(AuxC->r));
46
47
  int VerificaPontoInterno(const TPonto *AP, const TCirc *AC){
48
49
     float d, auxX, auxY;
50
51
    auxX = (AP->x - AC->centro.x);
52
    auxY = (AP->y - AC->centro.y);
53
    d = sqrt(auxX*auxX + auxY*auxY);
54
    if (d>AC->r)
55
       return 0;
56
     else
57
       return 1;
58
```

Correção: Valor da questão 3.0,

- (a) 10%,
- (b) 15%, deve usar a estrutura anterior
- (c) 20%, prototipo, passagem por referencia, leitura
- (d) 20%, prototipo, passagem por referencia, leitura
- (e) 20%, prototipo, passagem por referencia e dados constante, processamento
- (f) 15%, declaração de variáveis, chamadas a funções, uso do retorno.

Questão 3

```
#include <stdio.h>
2
  #include <math.h>
3
4
  typedef struct {
5
    int num;
6
    int den;
  }TRacional;
7
8
  void LeRacional(TRacional *);
10 TRacional SomaRacional (const TRacional *, const TRacional *);
  void PrnRacional(const TRacional *);
11
12
13 int main() {
14 TRacional R1, R2, S;
```

```
15
16
     LeRacional(&R1);
17
     LeRacional(&R2);
18
     S = SomaRacional(&R1, &R2);
19
     PrnRacional(&S);
20
21
     return 0;
22
  }
23
24
  void LeRacional(TRacional *A){
     printf("Digite numerador: ");
25
     scanf("%d", &(A->num));
26
     printf("Digite denominador: ");
27
28
     scanf("%d", &(A->den));
29
30
  TRacional SomaRacional (const TRacional *R1, const TRacional *R2) {
31
32
     TRacional Res;
33
     Res.num = R1->num*R2->den + R2->num*R1->den;
34
35
     Res.den = R1->den * R2->den;
36
37
     return Res;
38
  }
  void PrnRacional(const TRacional *A){
     printf("Resultado: %d/%d\n", A->num, A->den);
40
41
```

Correção: Valor da questão 2.0,

- (a) 15%,
- (b) 85%, Dados utilizados, funciones inclusas, protótipos e definição, passagem de parâmetros por referencia, uso de retorno, sintaxe

Questão 4

```
int verificaAnagrama(const char *, const char *);
2
3
  int verificaAnagrama(const char *s1, const char *s2){
4
    char *p1, *p2;
5
    int cont = 0;
6
7
     //Inicializa os ponteiros p1 e p2
8
    p1 = s1;
9
    p2 = s2;
10
     //verifica o tamanho de str2 e coloca p2 al final de str2
11
12
    while (*p2! = ' \ 0')
13
       p2++;
14
       cont++;
15
    p2--;
16
17
     //verifica a igualdade de p1 (ini->fim) e p2(fim->ini)
18
19
     //se encontrar dois caracteres diferentes, encerra a funcao
```

```
20
     //se chegar ao final de str1 ou str2 encerra o laço
21
     while (*p1! = ' \setminus 0' \&\& cont > 0){
22
        if (*p1!=*p2) return 0;
23
       p1++;
24
       p2--;
25
       cont --;
26
27
28
     //verifica se o laço foi encerrado pelas strins de tamanho diferente
     if(*p1!='\setminus 0' \mid \mid cont!=0)
29
       return 0;
30
31
     else
32
       return 1;
33 }
```

Correção: Valor da questão 2.5,

- (a) 10%,
- (b) 90%, Dados utilizados, algoritmo correto, funcionamento para strings de tamanho diferente, EXISTEM vários algoritmos