

ATIVIDADE DE FIXAÇÃO 1

```
1) void altera1(int **p, int *a)
    { **p = *a;
      *a = *a +50;
    }

void altera2(int **p, int *b)
    { *p = b;
      *b = *b +30;
    }

int main()
{
    int x,y, *px, *py;

    x = 10;
    y = x + 20;

    px = &x;
    py = &y;

    printf("x = %d, End. x = %p, px = %p, y = %d \n",x,&x, px,y);
    altera1(&px, &y);
    printf("x = %d, End. x = %p, px = %p, y = %d \n",x,&x, px,y);
    getchar();

    printf("y = %d, End. y = %p, py = %p, x = %d \n",y,&y, py,x);
    altera2(&py, &x);
    printf("y = %d, End. y = %p, py = %p, x = %d \n",y,&y, py,x);
    getchar();

    return(0);
}
```

Dado o código acima, responda as seguintes questões:

- (a) Qual a diferença entre px e x?
- (b) Qual a diferença entre px e py?
- (c) Quais são os valores impressos pelo primeiro printf?
- (d) O que muda do primeiro printf para o segundo?
- (e) Quais os valores impressos pelo terceiro print?
- (f) O que muda do terceiro para o quarto print?
- (g) Explique a diferença entre o altera1 e o altera2.

- 2) Faça o rastreamento do códigos a seguir e diga o que os mesmos fazem, dizendo qual a diferença entre eles.

```
void misterio1(char b[TAM], float *dec, int p,
int i)
{
    if(p < strlen(b))
    {
        if(b[p] == '1')
            *dec = *dec + pow(2,i);
        misterio1(b,dec,++p, -i);
    }
}
```

```
float misterio2(char b[TAM], float dec, int p,
int i)
{
    if(p < strlen(b))
    {
        dec = misterio2(b,dec,p+1,i-1);
        if(b[p] == '1')
            dec = dec + pow(2,i);
    }
    return(dec);
}
```

Obs.: A string b, contém 0s e 1s.

Obs.: Na chamada de misterio1 e de misterio2 dec e p devem ser 0 e i deve ser strlen(b) - 1.

Obs.: a função pow, calcula a potência de 2 elevado a i.

Obs.: O número da posição 0(zero) de b, será a posição strlen(b) -1 do número binário.

Obs.: rastrear código significa, fazer uma simulação com valores mostrando o que acontece no código.

- 3) O código a seguir deveria devolver um vetor contendo a intersecção entre os dois vetores de entrada ordenados, o código possui erros, localize-os e diga como corrigi-los. Depois reescreva sem usar recursividade.

```
void inter(int V1[TAM], int V2[TAM], int V3[TAM],int i, int j, int q1, int q2, int q3)
{
    if(i < q1 || j < q2)
    {
        if(V1[i] < V2[j])
            q3 = inter(V1,V2,V3, i, j+1,q1,q2,q3);
        else if(V1[i] > V2[j])
            inter(V1,V2,V3, i, j+1,q1,q2,q3);
        else {V3[q3] = V1[j];
            q3 = inter(V1,V2,V3,i+1,j+1,q1,q2,q3+1);
        }
    }
    return(q3);
}
```

Obs.: Os valores iniciais de i, j e q3 é 0.