

### Soluções para a Questão 3 da Prova I

Duas soluções para a questão 3 da Prova são apresentadas :

1. Uma solução para a Questão 3 da Prova I com direitos autorais do aluno Vilmar Dahmer
2. Uma outra solução

1.

```
/* Name: 31288_P1.cpp
   Copyright: CopyLeft
   Author: Vilmar Dahmer
   Date: 06/04/09 14:47
   Description: media harmonica de elementos da interseccao de 3 conjuntos
*/

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int m = 0, n = 0, anterior1 = 0, anterior2 = 1, atual, i, j, k, ultc1, ultc2, ultc3,
    ultintersec;
    int c1[100] = {0}, c2[100] = {0}, c3[100] = {0}, intersec[100] = {};
    float media, soma;

    // testes
    //int cc1[4] = {1,2,3,4}, ultcc1 = 4, cc2[3] = {2,3,4}, ultcc2 = 3, cc3[3] = {1,2,3},
    ultcc3 = 3;
    printf("MEDIA HARMONICA\n\n");

    while (m <= 0 || m > 20) {
        printf("Entre com termo 'm' (maior que 0, menor ou igual a 20): ");
        scanf("%d", &m);
    }

    while (n <= 0 || n > 40) {
        printf("Entre com termo 'n' (maior que 0, menor ou igual a 40): ");
        scanf("%d", &n);
    }

    // impares de m a n
    ultc1 = 0;
    printf("\n\tC1, impares de %d a %d: [ ", m, n);
    for (i = m; i <= n; i++) {
        if (i % 2 == 1) {
            c1[ultc1] = i;
            ultc1++;
            printf("%d ", i);
        }
    }
    printf("\n\n");
```

```

// multiplos de 3, de 1 a n
ultc2 = 0;
printf("\tC2, multiplos de 3, de 1 a %d: [ ", n);
for (i = 1; i <= n; i++) {
    if (i % 3 == 0) {
        c2[ultc2] = i;
        ultc2++;
        printf("%d ", i);
    }
}
printf("]\n\n");

// Fibonacci
atual = anterior2;
ultc3 = 1;
c3[0] = atual;
printf("\tC3, Fibonacci ate %d: [ %d ", m, atual);
for (i = 1; i < m; i++) {
    atual = anterior1 + anterior2;
    anterior1 = anterior2;
    anterior2 = atual;
    c3[ultc3] = atual;
    ultc3++;
    printf("%d ", atual);
}
printf("]\n\n");

// interseccao c1, c2, c3
printf("\tInterseccao C1, C2, C3: [ ");
ultintersec = 0;
for (i = 0; i < ultc1; i++) {
    for (j = 0; j <= ultc2; j++) {
        for (k = 0; k <= ultc3; k++) {
            //printf("\nc1=%d, c2=%d, c3=%d\n", c1[i], c2[j], c3[k]);
            if (c1[i] == c2[j] && c2[j] == c3[k]) {
                intersec[ultintersec] = c1[i];
                printf("%d ", intersec[ultintersec]);
                ultintersec++;
            }
        }
    }
}
printf("]\n\n");

// media harmonica
if (ultintersec > 0) {
    soma = 0.0;
    for (i = 0; i < ultintersec; i++) {
        soma += 1.0 / (float)intersec[i];
        //printf("elem: %d\n", intersec[i]);
    }
}

```

```

    }
    media = (float)ultintersec / soma;
    printf("Media harmonica de %d elementos: %.2f\n\n", ultintersec, media);
} else {
    printf("Nao existem elementos resultantes da interseccao para calculo de
media!\n\n");
}

#ifdef __WIN32__
    system("PAUSE");
#endif
return 0;
}

```

## 2. A outra....

```

//Dados 2 inteiros m, n, maiores que zero, m<=20 e n<=40,
//fazer a interseccao de 3 conjuntos:
//C1 = impares do intervalo [m,n] ; C2 = multiplos de 3, de 1 a n;
//C3 = numeros Fibonacci até o termo m.
// Mostrar a interseccao R e os 3 conjuntos gerados.
#include<stdio.h>
#include<stdlib.h>
#define MAX 40
#define MAX2 20
int main()
{
    int i, j, k, w; //indices
    int c1 [MAX], c2[MAX], c3[MAX2];
    // Vetores:  c1 - valores impares de [m,n];
    //           c2 - multiplos de 3 de [1,n];
    //           c3 - elementos Fibonacci ate m-esimo termo.
    int r[MAX], auxr[MAX]; // vetores intersecao e auxiliar de intersecao
    int m, n, limc1, limauxr, c; // valores limites a serem lidos e guardados
    double mh, soma;
    do // Leitura e consistencia dos valores de m, n
    {
        do
        {
            printf("\nDigite valor para m, maior que zero e menor que %d :", MAX2);
            scanf("%d",&m);
            printf("\n");
            if (m < 1 || m > MAX2)
                printf("Valor %d fora do intervalo",m);
        }
        while (m < 1 || m > MAX2);
        do
        {
            printf("\nDigite valor para n, maior que zero e menor que %d :", MAX);
            scanf("%d",&n);

```

```

        printf("\n");
        if (n < 1 || n > MAX)
            printf("Valor %d fora do intervalo",n);
    }
    while (n < 1 || n > MAX);
    if (m > n)
        printf("Valor %d maior que %d  Digitar novamente!",m,n);
    }
    while (m > n);
    system("cls");
    k = m; // geracao do vetor C1
    if (k % 2 == 0)
        k++;
    i=0;
    while (k <= n)
    {
        c1[i] = k;
        k = k +2;
        i = i +1;
    }
    printf("\n\n Vetor dos impares de %d a %d \n",m,n);
    for (j = 0; j < i; j++)
        printf("%4d",c1[j]);
    limc1 = i;
    // geracao do vetor C2
    for (i = 0; 3 * (i+1) <= n ; i++)
        c2[i]= 3 * (i+1);
    w = i ;
    printf("\n\n Vetor dos multiplos de 3 de 1 a %d \n",n);
    for (i = 0; i < w ; i++)
        printf("%4d",c2[i]);
    // geracao do vetor C3
    c3[0] = 1;
    c3[1] = 1;
    if (m >2)
        for (i = 2; i < m ; i++)
            c3[i] = c3[i-1] + c3[i -2];
    printf("\n\n Vetor Fibonacci ate o %d-esimo termo \n",m);
    for (i = 0; i < m ; i++)
        printf("%5d",c3[i]);
    // interseccao entre C1 e C2
    k = 0;
    for (i = 0; i < limc1 ; i++)
        for(j = 0; j < n ; j++)
            if ( c1[i] == c2[j])
            {
                auxr[k] = c1[i];
                k++;
            }
    }

```

```

// printf("\n\n Intersecao C1 e C2 tem %d elemento(s)****: \n\n",k);
// for (i=0; i < k ; i++)
//   printf("%4d",auxr[i]); // vetor primeira interseccao
// interseccao com c3
limauxr = k;
k = 0; c = 0;
for (i = 0; i < limauxr ;i++)
    for(j = 0; j < m ;j++)
        if (auxr[i] == c3[j])
            {
                r[k] = auxr[i];
                k++;
                c = 1;
            }
soma = 0;
if (c == 0)
{
    printf("\n\n Interseccao vazia, com estes vetores C1, C2 e C3 !!");
    printf("\n\n Impossivel calcular a media harmonica !!");
}
else
{
    printf("\n\n Interseccao C1 , C2 e C3 tem %d elemento(s): \n\n",k);
    for (i = 0; i < k ;i++)
        {
            printf("%4d",r[i]);
            soma = soma + float(1)/r[i];
        }
    mh = k / soma;
    printf("\n\n Media harmonica do(s) %d valor(es) da intersecao: %lf",k , mh);
}
printf("\n\n");
system("pause");
return 0;
}

```