

```

        }

        l = p[b][a] / p[a][a]; // Calcula o valor a ser aplicado as linhas

        for(c = a; c < n; c++)
        { // Aplica o valor calculado as linhas
            p[b][c] = p[b][c] - p[a][c]*l;
        }

        B[b] = B[b] - B[a]*l; // Aplica o valor calculado ao vetor B
    }
}

void pivotear(int n, double **p, double *B, int b, int a)
{
    double maior;
    int i; //Contador para o for
    int j; //Posição do maior elemento
    double *q, aux;

    maior = fabs(p[b][a]);
    j = b;

    for(i = b; i < n; i++)
    {
        if(fabs(p[i][a]) > maior)
        {
            maior = fabs(p[i][a]);
            j = i;
        }
    }

    //Troca as linhas para evitar divisão por 0
    q = p[b];
    p[b] = p[j];
    p[j] = q;

    aux = B[b];
    B[b] = B[j];
    B[j] = aux;
}

double *calculaResultado(int n, double **p, double *B) //Calcula o resultado para as variaveis da matriz
{
    double *result = calloc(n, sizeof(double));
    double sum;
    int a, b, count=(n-1);

    for(a = (n-1); a >= 0; a--)
    {
        sum = 0;

        for(b=count; b < n; b++)
        {
            sum += (p[a][b]*result[b]);
        }

        result[a] = (B[a]-sum)/p[a][a];
        count--;
    }

    return result;
}

void testaValores(int n, double *polinomio)
{
    char a;
    double x;
    double resultado;
    int i;
    getchar();
    printf("\n\nDeseja testar algum valor? s/n\n");
    a = getchar();
    while(a!='s')
    {
        scanf("%lf", &x);
        resultado = 0;
        for(i = 0; i < n; i++)
        {
            if(i==0) resultado += polinomio[i];
            else resultado += polinomio[i] * pow(x,i);
        }
        printf("\nResultado para f(%2.3lf): %5.5lf\n\n", x, resultado);
        getchar();
        printf("Deseja testar algum valor? s/n\n");
        a = getchar();
    }
}

```

I/O da execução do programa “Interpolação por sistema linear” (Linhas seguidas de ‘>’ se referem a entrada)

Quantos pontos serão lidos?

>3

x0:

>-1

f(x0):

>4

x1:

>0

f(x1):

>1

x2:

>2

f(x2):

>-1

Forma Vetorial:

$$P(-1.000) = A0*(-1.000)^0 + A1*(-1.000)^1 + A2*(-1.000)^2 = +4.000$$

$$P(+0.000) = A0*(+0.000)^0 + A1*(+0.000)^1 + A2*(+0.000)^2 = +1.000$$

$$P(+2.000) = A0*(+2.000)^0 + A1*(+2.000)^1 + A2*(+2.000)^2 = -1.000$$

Forma Matricial:

$$\begin{bmatrix} +1.000 & -1.000 & +1.000 \\ +1.000 & +0.000 & +0.000 \\ +1.000 & +2.000 & +4.000 \end{bmatrix} \begin{bmatrix} +4.000 \\ +1.000 \\ -1.000 \end{bmatrix}$$

Forma Matricial:

$$\begin{bmatrix} +1.000 & -1.000 & +1.000 \\ +0.000 & +3.000 & +3.000 \\ +0.000 & +0.000 & -2.000 \end{bmatrix} \begin{bmatrix} +4.000 \\ -5.000 \\ -1.333 \end{bmatrix}$$

Polinômio Interpolador:

$$P(x) = +0.667(x^2) -2.333x +1.000$$

Deseja testar algum valor? s/n

>s

>10

Resultado para f(10.000): 44.33333

Deseja testar algum valor? s/n

>s

>-1

Resultado para f(-1.000): 4.00000

Deseja testar algum valor? s/n

>n

```

void testaValores(int n, double *polinomio)
{
    char a;
    double x;
    double resultado;
    int i;
    getchar();
    printf("\n\nDeseja testar algum valor? s/n\n");
    a = getchar();
    while(a=='s')
    {

        scanf("%lf", &x);
        resultado = 0;
        for(i = 0; i < n; i++)
        {
            if(i==0) resultado += polinomio[i];
            else    resultado += polinomio[i] * pow(x,i);
        }

        printf("\nResultado para f(%2.3lf): %5.5lf\n\n", x, resultado);

        getchar();
        printf("Deseja testar algum valor? s/n\n");
        a = getchar();
    }
}

```

I/O da execução do programa “Interpolação por Lagrange” (Linhas seguidas de ‘>’ se referem a entrada)

Quantos pontos serão lidos?

>3

x0:

>-2

f(x0):

>2

x1:

>0

f(x1):

>-2

x2:

>4

f(x2):

>1

Polinômio Interpolador:

$P(x) = +0.458(x^2) - 1.083x - 2.000$

Deseja testar algum valor? s/n

>s

>10

Resultado para f(10.000): 33.00000

Deseja testar algum valor? s/n

>s

>4

Resultado para f(4.000): 1.00000

Deseja testar algum valor? s/n

>n

Interpolação por Newton

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//3 -1 4 0 1 2 -1

void multiplicaPolinomio(int n1, int n2, double *polinomio1, double *polinomio2);
void somaPolinomio(int n, double *polinomio1, double *polinomio2);
double *calculaResultado(int n, double *vetorx, double *vetorFx);
void lerPontos(int n, double *vetorx, double *vetorFx);
void imprimeResultado(int n, double *resultado);
void testaValores(int n, double *resultado);

int main()
{
    int n;
    double *vetorx, *vetorFx, *resultado;

    printf("Quantos pontos serão lidos?\n");
    scanf("%d", &n);

    vetorx = (double *) calloc(n,sizeof(double));
    vetorFx = (double *) calloc(n,sizeof(double));

    lerPontos(n,vetorx,vetorFx);
    putchar('\n');
    resultado = calculaResultado(n,vetorx,vetorFx);
    imprimeResultado(n,resultado);

    testaValores(n,resultado);

    free(vetorx); free(vetorFx); free(resultado);
}

void imprimeResultado(int n, double *resultado)
{ //Imprime o Polinomio interpolador encontrado
    int i;
    printf("Polinômio Interpolador:\nP(x)= ");
    for(i = n-1; i >= 0; i--)
    {
        if(i==0) printf("%+2.3lf\n", resultado[i]);
        else
            if(i==1) printf("%+2.3lfx ", resultado[i]);
            else    printf("%+2.3lf(x^%d) ", resultado[i], i);
    }
}

void multiplicaPolinomio(int n1, int n2, double *polinomio1, double *polinomio2) //Multiplica dois polinomios
{ //Resultado será devolvido no polinômio 1
    double resultado[n1+n2-1];
    int i,j;

    for(i = 0; i < (n1+n2-1); i++)
        resultado[i] = 0;

    for(i = 0; i < n1; i++)
        for(j = 0; j < n2; j++)
            resultado[i+j] += polinomio1[i] * polinomio2[j];

    for(i = 0; i < (n1+n2-1); ++i)
        polinomio1[i] = resultado[i];
}

void somaPolinomio(int n, double *polinomio1, double *polinomio2) //Soma dois polinomios
{ // Resultado será devolvido no polinômio 1
    int i;
    for(i = 0; i < n; ++i)
        polinomio1[i] += polinomio2[i];
}

double *calculaResultado(int n, double *vetorx, double *vetorFx) //Realiza todos os calculos referentes
{ //Ao método de newton e retorna o polinômio interpolador
    double m[n+n-1][n], aux[n+n-1], diferenca[n], polinomios[n][n], polinomio[2], a;
    double *resultado = calloc(n,sizeof(double));
    int auxiliar = n, flag = 1, i,j;
```

```

for(i = 0; i < n; i++)
{
    //inicializa aux e m
    aux[i*2] = vetorx[i];
    m[i*2][0] = vetorFx[i];
}

for(i = 1; i < n; i++)
    for (j = i; j < n+n-2; j+=2)
        //Realiza os calculos em m
        m[j][i] = m[j+1][i-1] - m[j-1][i-1];
        a = (aux[j+i])-(aux[j-i]);
        m[j][i]/=a;
}

for(i = 0; i < n; i++)//Inicializa o vetor diferenca
    diferenca[i] = m[i][i];

polinomios[0][0]=vetorFx[0];
polinomio[1] = 1;

for(i = 1; i < n; i++)
{
    //Realiza as multiplicações entre os polinomios
    polinomios[i][0] = -vetorx[0];
    polinomios[i][1] = 1;
    for(j = 1; j < i; j++)
    {
        polinomio[0] = -vetorx[j];
        multiplicaPolinomio(n,n,polinomios[i],polinomio);
    }
}

for (i = 1; i < n; i++)//Multiplica cada elemento do polinomio pelo valor da diferença
    for(j = 0; j < n; j++)
        polinomios[i][j] *= diferenca[i];

for (i = 0; i < n; i++)//Soma todos os polinomios, gerando o resultado final
    somaPolinomio(n,resultado,polinomios[i]);

return resultado;
}

void lerPontos(int n, double *vetorx, double *vetorFx)
{
    //Le pontos do console do usuário
    int i;
    for(i = 0; i < n; i++)
    {
        printf("x%d: ", i);
        scanf("%lf",vetorx+i);
        printf("f(x%d): ", i);
        scanf("%lf",vetorFx+i);
    }
}

void testaValores(int n, double *polinomio)
{
    //Testa valores inseridos pelo usuário, no polinomio
    char a;
    double x;
    double resultado;
    int i;
    getchar();
    printf("\n\nDeseja testar algum valor? s/n\n");
    a = getchar();
    while(a!='s')
    {
        scanf("%lf", &x);
        resultado = 0;
        for(i = 0; i < n; i++)
        {
            if(i==0) resultado += polinomio[i];
            else resultado += polinomio[i] * pow(x,i);
        }
        printf("\nResultado para f(%2.3lf): %5.5lf\n\n", x, resultado);
        getchar();
        printf("Deseja testar algum valor? s/n\n");
        a = getchar();
    }
}

```

I/O da execução do programa “Interpolação por Newton”
(Linhas seguidas de ‘>’ se referem a entrada)

Quantos pontos serão lidos?

>3

x0:

>-1

f(x0):

>4

x1:

>0

f(x1):

>1

x2:

>2

f(x2):

>-1

Polinômio Interpolador:

$P(x) = +0.667(x^2) - 2.333x + 1.000$

Deseja testar algum valor? s/n

>s

>0

Resultado para f(0.000): 1.00000

Deseja testar algum valor? s/n

>s

>2

Resultado para f(2.000): -1.00000

Deseja testar algum valor? s/n

>n