lista 02

01.c

```
#include <stdio.h>

void paridade(int n)
{
	printf( (n%2==0 ? "0\n" : "1\n" ));

}


int main()
{
	int n=0;
	scanf("%d", &n);
	paridade(n);
	return 0;
}
```

02.c

```
#include <stdio.h>



float volume(float n1, float n2, float n3)
{
	return n1 * n2 * n3;
}


int main()
{
	float n1, n2, n3, resultado;
	scanf("%f %f %f", &n1, &n2, &n3);
	resultado = volume(n1, n2, n3);
	printf("%.2f\n",resultado);
	return 0;
}
```

03.c

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

void pi_aprox(){
	double a1 = 0;
	double a = 1;
	double b = 1.0/sqrt(2);
	double t = 1.0/4.0;
	double p = 1.0;

	for (int i = 0; i < 25; ++i)
	{
		a1 = ( a+b )/2;
		b = sqrt( a*b );
		t = t - pow( p *( a - a1 ),2);
		p = 2*p;
		a = a1;
	}

	double pi = (a+b)* (a+b)/(4.0*t);

	printf("%lf\n", pi);


}


int main()
{
	pi_aprox();
	return 0;
}
```

04.c

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

float soma( float n1, float n2)
{
	return n1 + n2;
}

float sub( float n1, float n2)
{
	return n1 - n2;
}

float mult( float n1, float n2)
{
	return n1 * n2;
}

float div( float n1, float n2)
{
	return n1 / n2;
}

float exp02( float n1, float n2)
{
	return pow(n1 , n2);
}


float calc(int op, float n1, float n2)
{

		switch(op) {

	case 1  :
		return soma( n1, n2);
		break;

	case 2  :
		return sub( n1, n2);
		break;

	case 3  :
		return mult( n1, n2);
		break;

	case 4  :
		return div( n1, n2);
		break;

	case 5  :
		return exp02( n1, n2);
		break;
```

```
        }

        return 0;

}



int main()
{
        float n1, n2, resultado;
        int op;

        scanf("%d %f %f", &op, &n1, &n2);
        resultado = calc(op, n1, n2);
        printf("%f\n",resultado);
        return 0;
}
```

05.c

```
#include <stdio.h>


unsigned int fat(unsigned int n, unsigned int acc)
{
	if (!n){
		return acc;
	}

	return fat(n-1, acc*n);

}

unsigned int fatorial(unsigned int n)
{
	return fat(n, 1);

}

unsigned int coeficienteBinomial(unsigned int n, unsigned int k)
{

	return  fatorial(n)
			/fatorial(k)
			/fatorial(n-k);

}



int main()
{

	unsigned int n1, n2, resultado;

	scanf("%d %d", &n1, &n2);

	resultado = coeficienteBinomial(n1, n2);
	printf("%d\n",resultado);
	return 0;
}
```

06.c

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

double fDelta(double a, double b, double c)
{

	return (pow(b,2)-4*a*c);
}

void baskara(double a, double b, double c)
{
	double result1, result2, delta = fDelta(a,b,c);

	if (delta>0) {

		delta = sqrt(delta);
		result1 = (-b + delta)/ (2*a);
		result2 = (-b - delta)/ (2*a);
		printf("%lf %lf\n",result1, result2);

	}

	if (delta==0) {
		result1 = -b/ (2*a);
		printf("%lf\n",result1);

	}

	if (delta<0){
		delta = sqrt(-delta);
		result1 = (-b + delta)/ (2*a);
		result2 = (-b - delta)/ (2*a);
		printf("%lfi %lfi\n",result1, result2);

	}


}



int main()
{

	double a, b, c;


	scanf("%lf %lf %lf", &a, &b, &c);

	baskara(a, b, c);
	return 0;
}
```

07.c

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

float CtoK(float valor)
{
	return (valor + 273.15);
}

float FtoK(float valor)
{
	return ((valor + 459.67) * 5/9);
}

float KtoC(float valor)
{
	return (valor - 273.15);
}

float KtoF(float valor)
{
	return (valor * 9/5 - 459.67);
}

void temperatura(int uEntrada, int uSaida, float valor)
{

	if( uEntrada != 3){
		valor = (uEntrada == 1 ?  CtoK(valor) : FtoK(valor));
	}

	switch (uSaida)
	{
		case 1:
			printf("%.2f C\n", KtoC(valor));
			break;
		case 2:
			printf("%.2f F\n", KtoF(valor));
			break;
		case 3:
			printf("%.2f K\n", valor);
			break;

	}

}


int BtoD(int valor)
{

	int sum = 0;
	int i = 0;

	while(valor > 0)
```

```
	{
		sum += ((valor%10) == 0 ? 0 : pow( (valor%10)*2, i));
		valor /=10;
		i++;
	}
	return sum;
}

int DtoB(int valor)
{

	int i = 0;
	int sum = 0;
	while(valor > 0)
	{
		sum += (valor%2) * pow(10 , i++);
		valor /= 2;
	}

	sum += valor * pow(10 , i) ;
	return sum;
}


void numerico(int uEntrada, int uSaida, float valor){

	if (uEntrada == uSaida)
	{
		printf("%.0f\n", valor);
	}else if(uEntrada == 1)
	{
		printf("%d\n", DtoB(((int)valor)));
	}else
	{
		printf("%d\n", BtoD(((int)valor)));
	}

}

void converte(int conversao, int uEntrada, int uSaida, float valor)
{
	if (conversao==1){
		temperatura(uEntrada, uSaida, valor);
	}else{
		numerico(uEntrada, uSaida, valor);
	}
}

int main()
{
	int conversao, uEntrada, uSaida;
	float valor;
	scanf("%d %d %d %f", &conversao, &uEntrada, &uSaida, &valor);
	converte(conversao, uEntrada, uSaida, valor);
	return 0;
}
```

08.c

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



float salarioBase(int op)
{
	switch(op){
		case 1:
			return 10000;
			break;
		case 2:
			return 8000;
			break;
		case 3:
			return 5000;
			break;
		case 4:
			return 3000;
			break;
		case 5:
			return 2000;
			break;
	}

	return 0;
}

float descontos(float salario, int faltas)
{
	return salario/20*faltas;
}

float acrescimos(float salario, int horas)
{
	return (salario/(20*8) + 40) * horas;
}


float salario(int cargo, int faltas, int horas)
{
	float salario = salarioBase(cargo);
	return salario - descontos(salario, faltas) + acrescimos(salario, horas);
}


int main()
{
	int cargo, faltas, horas;
	scanf("%d %d %d", &cargo, &faltas, &horas);
	printf("%.2f\n", salario(cargo, faltas, horas));
	return 0;
}
```

09.c

```
#include <stdio.h>



int expo(int x, int y)
{
	if ( y == 2) return x*x;
	if ( y == 1) return x;
	if ( y == 0) return 1;

	int expA = expo(x, y/2);
	int expB = expo(x, y%2);

	return expA*expA*expB;

}


int main()
{
	int x, y;
	scanf("%d %d", &x, &y);
	printf("%d\n", expo(x, y));
	return 0;
}
```

10.c

```
#include <stdio.h>

int fibMen[1000];

int fib (int n){

	if(!n) return 0;
	if(fibMen[n]) return fibMen[n];

	fibMen[n] = fib(n-1) + fib(n-2);

	return fibMen[n];

}


int main()
{
	int n;
	fibMen[0] = 0;
	fibMen[1] = 1;
	fibMen[2] = 1;

	scanf("%d", &n);
	fib(n);
	for (int i = 0; i <= n; ++i)
	{
		printf("%d ", fibMen[i]);
	}

	printf("\n");


	return 0;
}
```

11.c

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

int BtoD(int valor, int sum, int i)
{
	if (valor <= 0) return sum;

	sum += ((valor%10) == 0 ? 0 : pow( (valor%10)*2, i));
	valor /=10;
	i++;

	return BtoD(valor, sum, i);
}

int DtoB(int valor, int sum, int i)
{

	if (valor <=0) return sum + valor * pow(10 , i) ;

	sum += (valor%2) * pow(10 , i++);
	valor /= 2;


	return DtoB(valor, sum, i);
}

int converte(int valor, int entrada){

	if (entrada == 1)
	{
		return DtoB(valor, 0, 0);
	}else if(entrada == 2)
	{
		return BtoD(valor, 0, 0);
	}

	return -1;

}
int main()
{
	int valor, entrada;

	scanf("%d %d", &valor, &entrada);

	int saida = converte(valor, entrada);

	printf("%d\n", saida);

	return 0;
}
```

12.c

```c
#include <stdio.h>

void proxima( int anterior[], int atual[],  int tam, int linha)
{

	if (linha == tam) return;

	atual[0] = anterior[0];
	atual[tam] = anterior[tam-1];


	for (int i = 1; i < tam ; i++)
	{
		atual[i] = anterior[i-1] + anterior[i];
	}

	tam++;


	proxima(atual, anterior, tam, linha);

}


int pascPiramide ( int linha, int coluna){
	int impar[linha];
	int par[linha];
	int tam = 1;
	impar[0] = 1;
	if(linha>1)
	{
		proxima(impar, par, tam, linha);
	}

	if(linha%2==0)
	{
		return par[coluna];
	}
	return impar[coluna];

}
int main()
{

	int n, k;
	scanf("%d %d", &n, &k);

	int bi = pascPiramide(n+1, k);

	printf("%d\n", bi);


	return 0;
}
```

13.c

```
#include <stdio.h>


int MDC (int a, int b)
{
	return (b==0)? a : MDC(b, a % b);
}


int main()
{

	int a, b;
	scanf("%d %d", &a, &b);

	printf("%d\n", MDC(a,b));

}
```

14.c

```
#include <stdio.h>
#include <math.h>
int palindromoC(unsigned n, int dig)
{
    if (n < 10) return 1;

    int first = n/ pow(10,dig-1);
    int last = n%10;
    n -= first*pow(10,dig-1);
    n /= 10;

    if (first == last){
        return palindromoC(n, dig-2);

    }
    return -1;
}

int palindromo(unsigned int n)
{
    int dig;
    int aux = n;
    while(aux>0)
    {
        dig++;
        aux /=10;
    }
    return palindromoC(n, dig);
}




int main()
{
    unsigned int n;
    scanf("%d", &n);
    printf((palindromo(n)==1? "sim" : "nao"));
    printf("\n");
}
```

15.c

```c
#include <stdio.h>



void move(int n, char origem, char destino, char using)
{
	if (n==1){
		printf("%c-%c, ", origem, destino);
	}else{
		move(n-1, origem, using, destino);
		move(1, origem, destino, using);
		move(n-1, using, destino, origem);
	}

}

int main()
{
	int n;
	scanf("%d", &n);
	move(n, 'A', 'C', 'B');
	printf("\n");
	return 0;
}
```