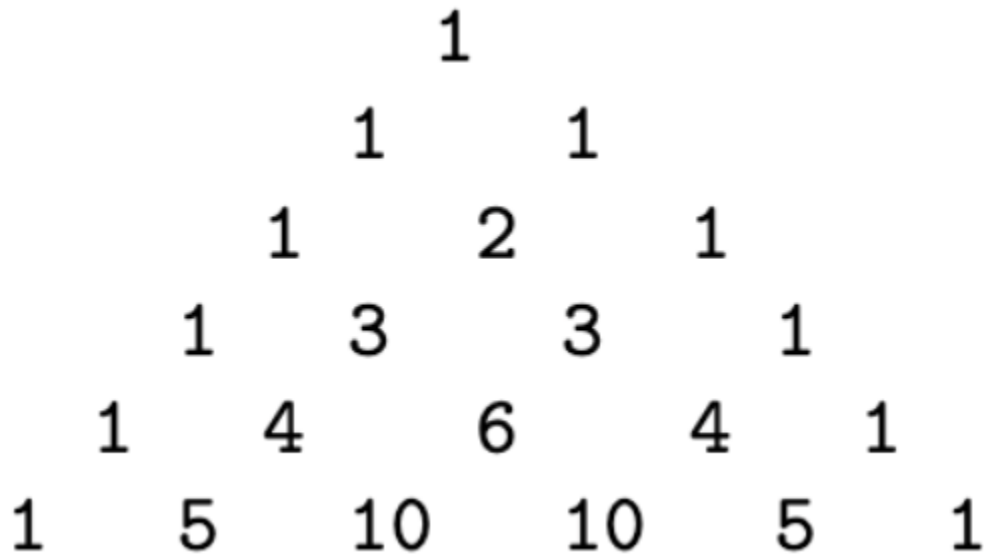


## Ficha 4

O objetivo principal desta ficha é familiarizar o aluno com a utilização e manipulação de arrays em C

### Arrays

1. Dado um vetor que armazena 6 números inteiros, o programa deverá executar os seguintes passos
  - a. Atribua os seguintes valores a esse vetor: 1, 0, 5, -2, -5, 7.
  - b. Armazena numa variável inteira a soma entre os valores nas posições 0, 1 e 5 do vetor e mostrar no ecrã essa soma.
  - c. Modifica o vetor na posição 4, atribuindo a esta posição o valor 100
  - d. Mostra no ecrã cada valor deste array, um em cada linha.
2. Crie um programa que lê 5 valores inteiros e, em seguida, mostre-os no ecrã na ordem inversa a que foram escritos.
3. Crie um programa que preencha um array de tamanho 100 com os 100 primeiros naturais que não são múltiplos de 7 ou que terminam com 7.
4. Dado um conjunto de notas dos N alunos de uma disciplina, calcular o número de notas superiores à nota média dessa disciplina.
5. Dados os valores mensais da produção anual duma fábrica de automóveis, calcular:
  - a. os valores produzidos até cada mês, desde o início do ano (produção acumulada);
  - b. a produção total no final do ano;
  - c. o mês na qual foi alcançada metade da produção anual
6. Dada uma sequência de N números inteiros, procurar a ocorrência de um determinado valor x nessa sequência.
7. Dada uma sequência de números inteiros não negativos, ordenar essa sequência por ordem decrescente.
8. Defina uma função `void swap(int v[], int i, int j)` que troca o valor das posições i e j do array v.
9. Defina uma função `int removeRep( int v[], int n)` que remove do array v todos os elementos repetidos (deixando apenas uma cópia). O parâmetro n corresponde ao número de elementos do array e a função deverá retornar o número de elementos com que o array ficou.
10. O triângulo de Pascal é uma forma de calcular os coeficientes da expansão do binómio de Newton. Abaixo, relembramos as 5 primeiras linhas. Note-se que a linha n do triângulo tem n elementos e que a linha n+1 pode ser obtida apartir da linha n usando o seguinte processo:
  - acrescenta-se um 1 no final, i.e., coloca-se 1 na posição n
  - para todos os elementos (desde n-1 até 1, por esta ordem) substitui-se o elemento nessa posição pela sua soma com o que está na posição anterior.



11. Defina uma função `void insere ( int v[], int N, int a)` que insere um elemento a num array ordenado. Assuma que as N primeiras posições do array estão ordenadas e que por isso, após a inserção o array terá as primeiras N+1 posições ordenadas.
12. Usando a função anterior, podemos definir uma função de ordenação de um array:

```
void iSort( int v[], int N){
    int i;
    for (i = 1; i<N;i++)
        insere(v,i,v[i]);
}
```

Apresente uma definição alternativa deste algoritmo sem usar a função `insere`

## 'Strings'

1. Crie um programa que leia uma 'string' do ecrã e a imprima
2. Crie um programa que calcula o comprimento duma 'string' (não usar a `strlen`)
3. Defina uma função `void strrev(char s[])` que inverte uma 'string'
4. Defina uma função `void strnoV(char s[])` que retira todas as vogais de uma 'string'
5. Escreva uma função `void capitalizar(char str[])` que transforma todas as letras duma cadeia em maiúsculas; outros caracteres devem ficar inalterados. O argumento é uma 'string'.  
**Sugestão:** use a função `toupper` da biblioteca padrão para transformar cada carácter em maiúscula.
6. Escreva uma função `int scrabble(char str[])` que calcula a pontuação de uma palavra dada como cadeia de caracteres; pode assumir que a cadeia é constituída apenas por letras maiúsculas sem acentos. Recorde que a pontuação de palavras em inglês no jogo SCRABBLE:
  - A,E,I,L,N,O,R,T,S,U 1 ponto;
  - D,G 2 pontos;
  - B,C,M,P 3 pontos;
  - F,H,V,W,Y 4 pontos;
  - K 5 pontos;
  - J,X 8 pontos;
  - Q,Z 10 pontos.

A pontuação duma palavra é a soma dos pontos de letras individuais.  
Exemplo: PITFALL vale  $3 + 1 + 1 + 4 + 1 + 1 + 1 = 12$  pontos, logo `scrabble("PITFALL")` dá resultado 12.
7. Escreva uma função `int calc(char str[])` que implementa uma mini-calculadora: a string dada tem sempre 3 caracteres correspondentes a dois algarismos decimais ('0' até '9') e um sinal de operação no meio ('+', '-', '\*'). A função deve calcular o valor da expressão e retornar o inteiro correspondente. Exemplos: `calc("5-3")` dá 2 e `calc("2*3")` dá 6.
8. Defina uma função `void eliminar (char str[], char ch)` que elimina a primeira ocorrência de um carácter `ch` de uma string. Exemplo: se `str = "ABBA"`, então depois de executar `eliminar(str, 'B')` devemos ter `str = "ABA"`.

## funções da biblioteca-padrão

```
#include <ctype.h>
int isupper(int ch); // testes
int islower(int ch);
int isdigit(int ch);
int toupper(int ch); // converter em maiúsculas
int tolower(int ch); // converter em minúsculas
#include <string.h>
size_t strlen(char str[]); /* comprimento */
```