Análise Combinatória com o Interpretador Hall

O *interpretador Hall* disponibiliza as seguintes funções para se trabalhar com os conceitos da análise combinatória. As funções são:

- Permutacao(n)
- Fatorial(n)
- Arranjo(n,p)
- Combinacao(n,p)
- Binomial(n,p)
- Permutar('a', 'b', 'c',...)
- Arranjar(n, p, 'a', 'b', 'c',...)
- Combinar(n, p, 'a', 'b', 'c',...)

As funções **Permutacao(), Fatorial(), Arranjo(), Combinacao() e Binomial()** são funções numéricas do tipo inteiro, isto é, elas retornam valores do tipo inteiro. Os argumentos **n** e **p** devem ser inteiros positivos e **n >= p**. Pelo fato dessas funções serem numéricas, a utilização clássica delas é na atribuição de valores a variáveis. Veja os exemplos abaixo:

- x := Permutacao(5)
- y := Fatorial(7)
- w := Arranjo(5,2)
- z := Combinacao(4,3)
- u := Binomial(7,3)

Um exemplo de código pode ser visto abaixo no algoritmo

Funções da Análise Combinatória

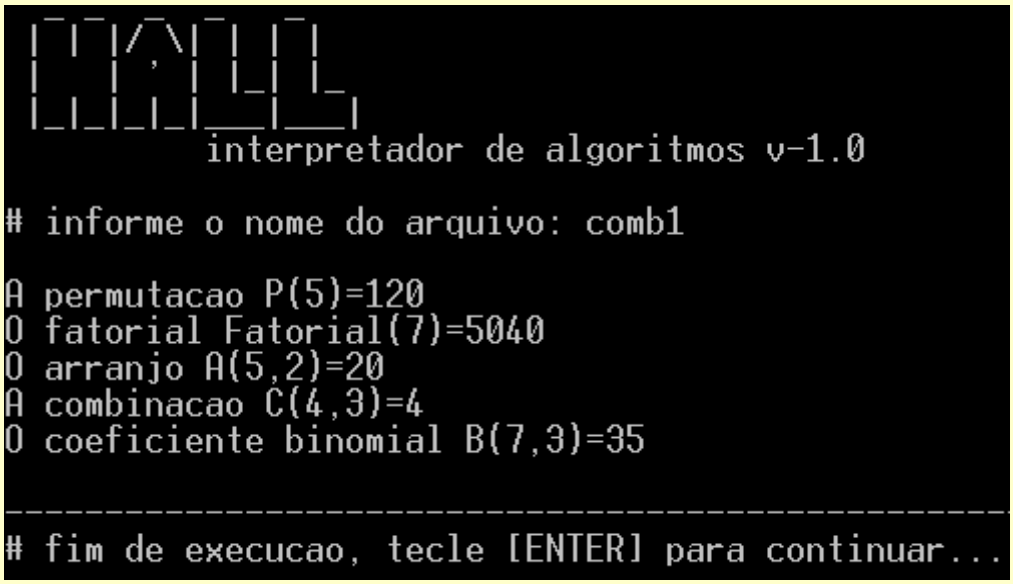
```
algoritmo()
{
	inteiro u,x,y,w,z;

	x := Permutacao(5);
	y := Fatorial(7);
	w := Arranjo(5,2);
	z := Combinacao(4,3);
	u := Binomial(7,3);

	escreva("A permutacao P(5)=",x);
	escreva("O fatorial Fatorial(7)=",y);
	escreva("O arranjo A(5,2)=",w);
	escreva("A combinacao C(4,3)=",z);
	escreva("O coeficiente binomial B(7,3)=",u);
}
```

A tela de execução do algoritmo acima pode ser vista na figura abaixo:

Execução do algoritmo



Criando as permutações, arranjos e combinações

As outras funções, a saber:

- Permutar('a', 'b', 'c',...)
- Arranjar(n, p, 'a', 'b', 'c',...)
- Combinar(n, p, 'a', 'b', 'c',...)

realizam as operações características de cada uma delas com os caracteres que são passados como argumentos.

Vejamos..

A função **Permutar()**, como o próprio nome indica, faz a permutação sem repetição dos caracteres passados como argumento. A função não exibe as possíveis permutações mas, retorna uma cadeia de caracteres separadas por vírgulas com essas permutações.

Vejamos um exemplo...

Vamos ver quantas são e quais saõ as permutações que a palavra “AMOR” nos permite fazer. Note que a palavra AMOR possui 4 letras, assim, a quantidade total de permutações será dado por Permutação de 4, isto é Permutacao(4). Para se fazer a permutação das letras da palavra AMOR, basta chamar a função Permutar() com a seguinte sintaxe: Permutar('A','M','O','R').

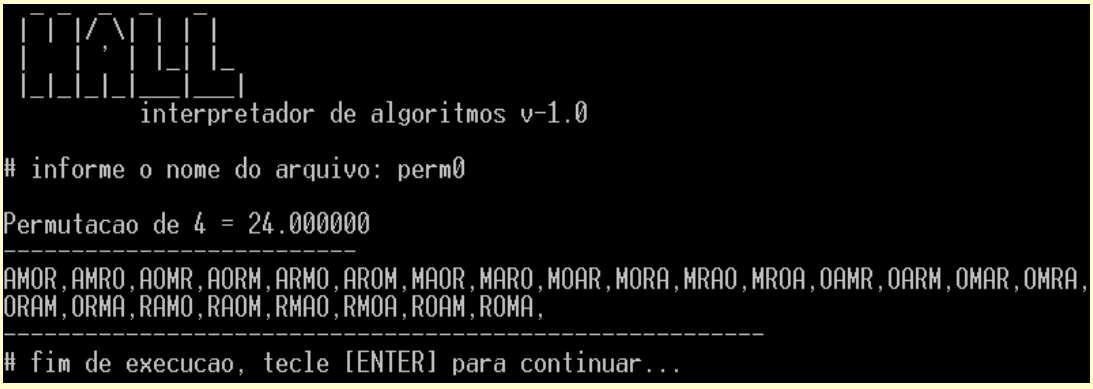
Análise Combinatória: Função Permutar()

```
algoritmo()
{
	cadeia x;

	escreva("Permutacao de 4 = ", Permutacao(4));
	escreva("-------------------------");
	x := permutar('A','M','O','R');
	escreva(x);
}
```

A tela de execução do exemplo acima pode ser visualizada na figura abaixo:

Função Permutar(): tela de execução



A função Combinar()

A função **Combinar()** exibe a combinação dos caracteres propriamente dita. Ela recebe três tipos de argumentos. Os dois primeiros, **n** e **p**, devem ser do tipo inteiro positivo e **n >= p**. Em seguida, deverá ser informado os caracteres sobre os quais a função irá atuar. Deve-se observar que a quantidade de caracteres informados deverá ser igual ao valor de **n**.

Abaixo podemos observar um exemplo de código que usa as funções acima e em seguida é apresentada a tela de execução do mesmo.

Veja...

Análise Combinatória: Função Combinacao()

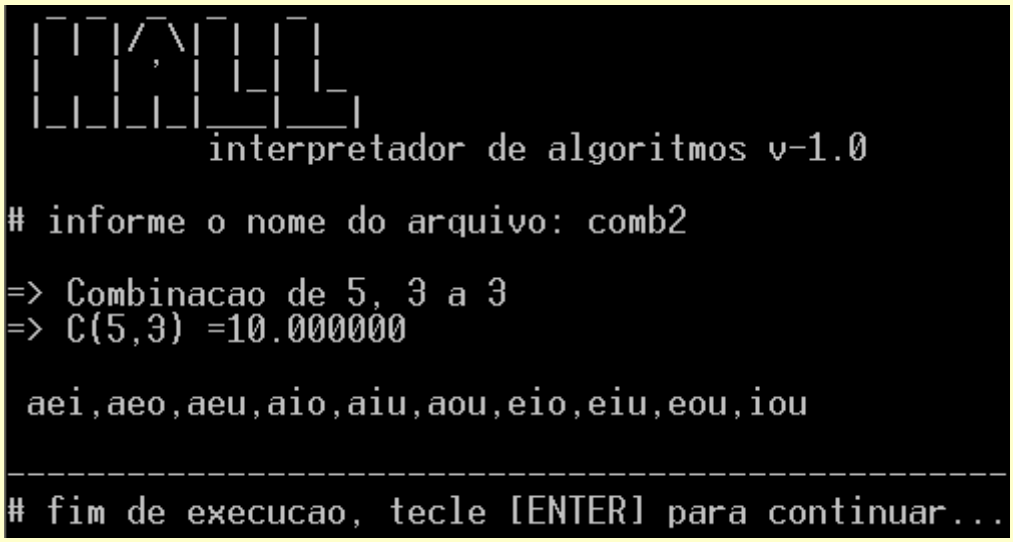
```
algoritmo()
{
	// exibe o valor e as combinacoes de 5, 3 a 3

	cadeia x;

	escreva("=> Combinacao de 5, 3 a 3");
	escreva("=> C(5,3) =", Combinacao(5,3));
	escreva();
	x := combinar(5,3,'a','e','i','o','u');
	escreva(x);
}
```

Veja abaixo a tela de execução desse algoritmo...

Função Combinacao(): tela de execução



Observação:

Na especificação dos caracteres a serem combinados, pode-se também, especificar os mesmos entre aspas.

Veja o exemplo abaixo...

Análise Combinatória: Função Combinacao()

```
algoritmo()
{
        // exibe o valor e as combinacoes de 5, 3 a 3

        cadeia x;

        escreva("=> Combinacao de 5, 3 a 3");
        escreva("=> C(5,3) =", Combinacao(5,3));
        escreva();
        x := combinar(5,3,”a”,”e”,”i”,”o”,”u”);
        escreva(x);
}
```

O resultado da execução é exatamente o mesmo do exemplo anterior.

Há ainda uma outra forma de se chamar a função **Combinar()**. Essa terceira forma usa variáveis.

Veja o exemplo abaixo:

Análise Combinatória: Função Combinacao()

```
algoritmo()
{
        // exibe o valor e as combinacoes de 5 3 a 3

        caracter a,e,i,o,u;
        cadeia x;

        a := 'a';
        e := 'e';
        i := 'i';
        o := 'o';
        u := 'u';

        x := combinar(5,3,a,e,i,o,u);
        escreva("Combinacao de 5, 3 a 3 = ",Combinacao(5,3));
        escreva(x);

        // observe que, na chamada abaixo
        // a,e,i,o e u sao variaveis que
        // coincidentemente contem as vogais
}
```

A função Arranjar()

A função **Arranjar()** exibe o arranjo dos caracteres propriamente dita. Ela recebe três tipos de argumentos. Os dois primeiros, **n** e **p**, devem ser do tipo inteiro positivo e **n >= p**. Em seguida, deverá ser informado os caracteres sobre os quais a função irá atuar. Deve-se observar que a quantidade de caracteres informados deverá ser igual ao valor de **n**.

Abaixo podemos observar um exemplo de código que usa as funções acima e em seguida é apresentada a tela de execução do mesmo.

Veja...

Análise Combinatória: Função Arranjo()

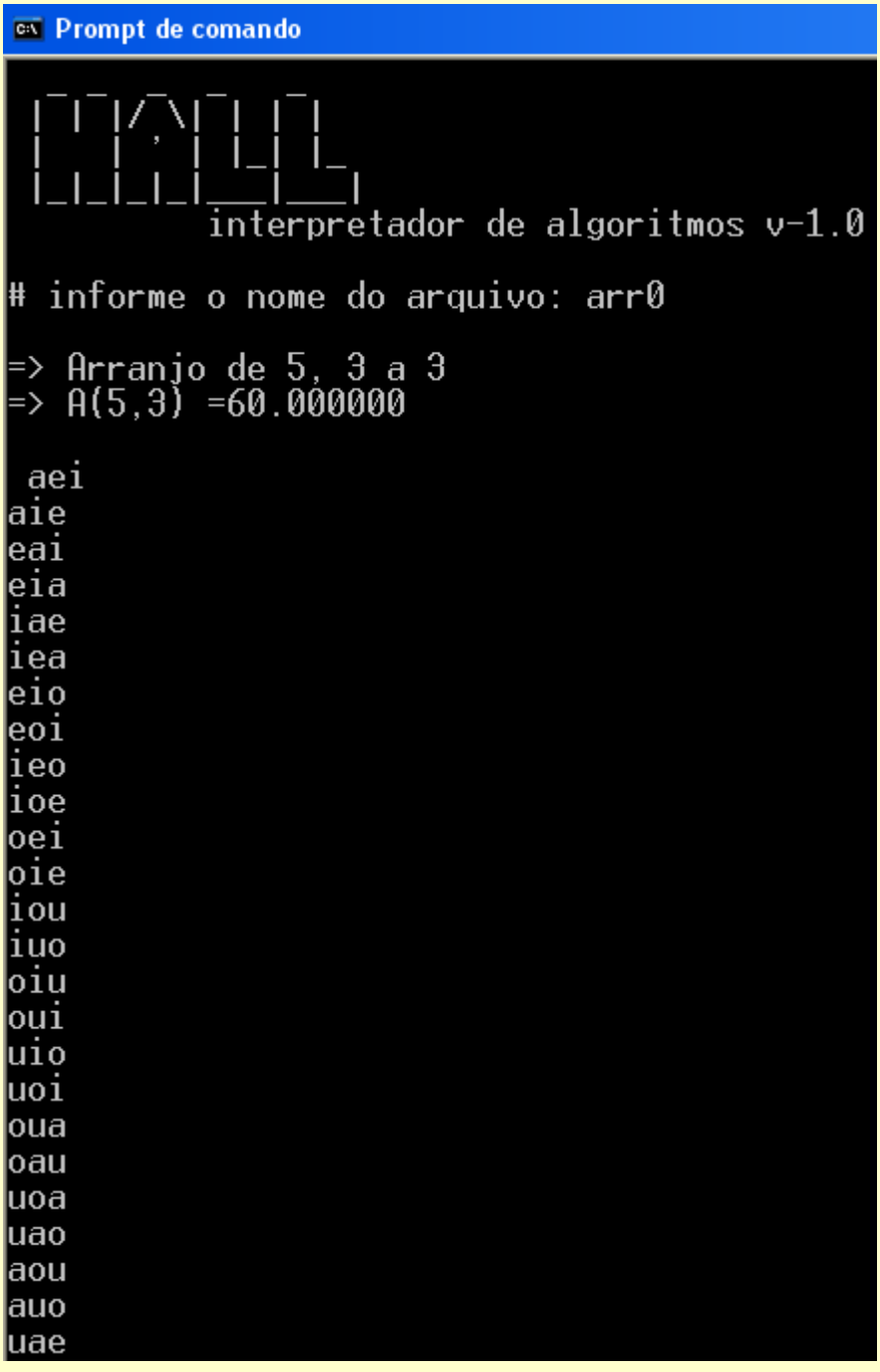
```
algoritmo()
{
	// exibe o valor e os arranjos de 5, 3 a 3

	cadeia x;

	escreva("=> Arranjo de 5, 3 a 3");
	escreva("=> A(5,3) =", Arranjo(5,3));
	escreva();
	x := Arranjar(5,3,'a','e','i','o','u');
	escreva(x);
}
```

Abaixo podemos ver a tela de execução do algoritmo...

Função Arranjo(): tela de execução



Ok! por hora é isso aí

Até,

fernandopaim@paim.pro.br