

solução do exame de PI de 17/01/2005

**Pergunta 1**

Objectivo primário de um compilador:

- Traduzir ó código legível pelos humanos em código executável pelo computador.

Além disso:

- Verificar os erros.

- Permitir escrever código independente da plataforma.

**Pergunta 2**

```
// versao 1: solucao com arrays
```

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

```
void main()
```

```
{
```

```
    int p[20], q[20], n, i, j;
```

```
    printf("numero de linhas:");
```

```
    scanf("%d", &n);
```

```
    for (i=0; i<20; i++)          // inicializar as variaveis
```

```
        p[i] = 0;
```

```
    p[1] = 1;
```

```
    for (i=1; i<=n; i++)          // i = linha actual
```

```
    {
```

```
        for (j=1; j<=i; j++)      // mostra linha actual
```

```
            printf("%d ", p[j]);
```

```
        printf("\n");
```

```
        for (j=1; j<=i+1; j++)    // calcula nova linha
```

```
            q[j] = p[j]+p[j-1];
```

```
        for (j=1; j<=i+1; j++)    // copia linha nova para
```

```
            p[j] = q[j];          // linha actual
```

```
    }
```

```
}
```

```
-----  
// versao 2: solucao com combinacoes
```

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

```
int factorial(int n)
```

```
{
```

```
    int i;
```

```
    if (n==0)
```

```
        return(1);
```

```
    else
```

```
        return(n*factorial(n-1));
```

```
}
```

```
int combin(int m, int n)
```

```
{
```

```
    return(factorial(m)/(factorial(n)*factorial(m-n)));
```

```
}
```

```

void main()
{
    int n, i, j;

    printf("numero de linhas:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        for (j=0; j<=i-1; j++)
            printf("%d ", combin(i-1, j));
        printf("\n");
    }
}

```

### Pergunta 3

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

```

void main()
{
    int n, i, j;
    char s[80];

    do {
        printf("palavra:");
        scanf("%s", s);
        if (strcmp(s,"sair"))
            printf("length = %d\n", strlen(s));
    } while (strcmp(s,"sair"));
}

```

### Pergunta 4

a)

```
typedef int matriz[3][3];
```

b)

```

void mult(matriz a, matriz b, matriz c)
// a notar: arrays sempre passagem por referencia
// entao mudancas a matriz c sao permanentes
{
    int i, j, k;
    for (i=0; i<3; i++)
        for (j=0; j<3; j++)
            {
                c[i][j]=0;
                for (k=0; k<3; k++)
                    c[i][j] += a[i][k]*b[k][j];
            }
}

```

c)

```

int gandalfiana(matriz a)
{

```

```

int i, j, soma, gndf=1;

for (i=0; i<3; i++)
{
    soma=0;
    for (j=0; j<3; j++)
        soma += a[j][i];
    if ((soma % 2)==0)
        gndf=0;
}
return(gndf);
}

```

### Pergunta 5

- a) A função FunN () retorna o número de dígitos do argumento.  
b) FunN(215) ----> 3

### Pergunta 6

a)

```

typedef struct {
    char marca[20];
    char modelo[20];
    int cilindrade;
    float preco;
} carro;

```

b)

```

carro carros[2];
strcpy(carros[1].marca, "Renault");
strcpy(carros[1].modelo, "Megane");
carros[1].cilindrade = 1400;
carros[1].preco = 17696.0;
strcpy(carros[2].marca, "Volkswagen");
strcpy(carros[2].modelo, "Golf");
carros[2].cilindrade = 1400;
carros[2].preco = 20104.0;

```

### Pergunta 6

Variável global: o âmbito é o programa inteiro

Variável local: tem âmbito limitado. Por exemplo apenas pode ser usada dentro de uma função.

Parâmetro: É uma variável local que recebe um valor na altura da chamada da função.