Lista de Tabelas

Lista de Listagens

1 Tabelas

tipo	bits	min...max	precisao
char	8	0..127	2
signed char	8	-128..127	2
unsigned char	8	0..255	2
short	16	-32.768 .. 32.767	4
unsigned short	16	0 .. 65.535	4
int	32	-2x10**9 .. 2 x 10**9	9
unsigned int	32	0 .. 4x10**9	9
int64_t	64	-9 x 10**18 .. 9 x 10**18	18
uint64_t	64	0 .. 18 x 10**18	19

Tabela 1: Limites de representação de dados

2 Codigos

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <math.h>
5
6  #include <inttypes.h>
7  #include <ctype.h>
8
9  #include <algorithm>
10 #include <utility>
```

0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5.040
8! = 40.320
9! = 362.880
10! = 3.628.800
11! = 39.916.800
12! = 479.001.600 [limite do (unsigned) int]
13! = 6.227.020.800
14! = 87.178.291.200
15! = 1.307.674.368.000
16! = 20.922.789.888.000
17! = 355.687.428.096.000
18! = 6.402.373.705.728.000
19! = 121.645.100.408.832.000
20! = 2.432.902.008.176.640.000 [limite do (u)int64_t]

Tabela 2: Fatorial

Tipe	%
char	c
int	d
float	e, E, f, g, G
int (octal)	o
int (hexa)	x, X
uint	u
char*	s

Tabela 3: scanf() - %[*][width][modifiers]type

modifiers	tipo
h	short int (d, i, n), or unsigned short int (o, u, x)
l	long int (d, i, n), or unsigned long int (o, u, x), or double (e, f, g)
L	long double (e, f, g)

Tabela 4: scanf() %[*][width][modifiers]type

```cpp
#include <iostream>

#include <map>
#include <set>
#include <vector>
#include <sstream>

using namespace std;

#define abs(a) ((a) > 0 ? (a) : -(a))

int main()
{
    int n;

    cin >> n;

    for (int i = 0; i < n; i++)
    {

    }

    while (cin >> n)
    {

    }
    return 0;
}
```

Código 1: Modelo

```cpp
const double EPS = 1e-10;
/**
 * -1 se x < y
 * 0 se x = y
 * 1 se x > y
 */
inline int cmp (double x, double y = 0, double tol = EPS)
{
    return (x <= y + tol) ? (x + tol < y) ? -1 : 0 : 1;
}
```

Código 2: comparcao de ponto flutuante

```
set ai noet ts=4 sw=4 bs=2
syn on
mat Keyword "\<foreach\>"
```

Código 3: .vimrc para a configuração do vim

```cpp
/* printf example */
#include <stdio.h>

int main()
{
    printf ("Characters: %c %c \n", 'a', 65);
    printf ("Decimals: %d %ld\n", 1977, 650000L);
    printf ("Preceding with blanks: %10d \n", 1977);
    printf ("Preceding with zeros: %010d \n", 1977);
    printf ("Some different radixes: %d %x %o %#x %#o \n", 100, 100, 100,
        100, 100);
    printf ("floats: %4.2f %+.0e %E \n", 3.1416, 3.1416, 3.1416);
    printf ("Width trick: %*d \n", 5, 10);
    printf ("%s \n", "A string");
    return 0;
}
/* output
Characters: a A
Decimals: 1977 650000
Preceding with blanks:       1977
Preceding with zeros: 0000001977
Some different radixes: 100 64 144 0x64 0144
floats: 3.14 +3e+000 3.141600E+000
Width trick:    10
A string
*/
```

Código 4: printf

```cpp
#include <iostream>
#include <queue>
using namespace std;

int main ()
{
  priority_queue<int> mypq;

  mypq.push(30);
  mypq.push(100);
  mypq.push(25);
  mypq.push(40);

  cout << "size: " << (int) mypq.size() << endl;

  cout << "Popping out elements...";
  while (!mypq.empty())
  {
     cout << " " << mypq.top();
     mypq.pop();
  }
  cout << endl;

  return 0;
}
```

Código 5: exemplo de priority queue

```cpp
#include <iostream>
#include <set>
using namespace std;

```

```
5  int main ()
6  {
7    multiset<int> mymultiset;
8    multiset<int>::iterator it;
9
10   // set some initial values:
11   for (int i=1; i<=5; i++) mymultiset.insert(i*10);   // 10 20 30 40 50
12
13   cout << "size: " << (int) mymultiset.size() << endl;
14
15   it=mymultiset.find(20);
16   mymultiset.erase (it);
17
18   if (! mymultiset.empty)
19    mymultiset.erase (mymultiset.find(40));
20
21   for (it=mymultiset.begin(); it!=mymultiset.end(); it++)
22     cout << " " << *it;
23
24   int myints[]={19,72,4,36,20,20};
25   multiset<int> first (myints,myints+3);      // 4,19,72
26   multiset<int> second (myints+3,myints+6);   // 20,20,36
27   multiset<int>::iterator it;
28
29   first.swap(second); // troca conteudo. o primeiro fica [20,20,36] e o
         segundo [4,19,72]
30
31   return 0;
32 }
```

Código 6: exemplo de multset