

```

/*
Programma : cppVerifical_4.cpp
Autore : Riontione Raffaele 4° informatici
        ITIS Molinari - Milano corso serale
        29/10/'10
Traccia : Scrivere un programma che ordina n numeri casuali
          con il metodo bubblesort, in modo che due elementi
          del vettore vengano scambiati con l'utilizzo dei puntatori.

aggiornamenti : 1/11/2010 - sistemata la funzione readInt.
aggiornamento : 2/11/2010 - errore cancellazione nella funz
*/
#include <iostream>
#include <stdlib.h>
#include <conio.h>
using namespace std;

void sort(int *&,int); //funzione bubblesort : riordina i dati

int readInt(char[]); //funzione che restituisce un intero

main()
{
    srand((unsigned)time(NULL)); //inizializzazione numeri casuali
    int i,dim = 0; //indice del for e dimensione array di numeri
    do{
        dim = readInt("quanti numeri casuali vuoi ordinare (da 2 a 100)? : ");
    }while(dim < 2 || dim > 100); //se la dimensione è minore di 2 ripete
    int *p = new int[dim]; //assegnazione di una array dinamica al puntatore
    cout << "\nGenerazione di numeri casuali tra 0 e 100\n\n";
    for (i = 0 ; i < dim ; i++)
    {
        *(p+i) = rand()%101;
        cout << "dato[" << i << "] = " << *(p+i) << endl;
    }
    sort(p,dim); //chiamata alla funzione sort per il riordino
    system("pause");
    cout << "\n\nriordino dei numeri :\n\n";
    for (i = 0 ; i < dim ; i++)
        cout << "dato[" << i << "] = " << *(p+i) << endl;
    delete p;
    system("pause");
} //chiusura main

//funzione sort : riordino dell'array
void sort(int *&p,int dim)
{
    int flag,k,temp,i;
    flag=1;
    k=dim-1;
    while ((flag) && (k > 0))
    {
        flag=0;
        for (i=0 ; i<k ; i++)
        {
            if (*(p+i) > *(p+i+1))

```

```

        {
            temp=*(p+i);
            *(p+i) = *(p+i+1);
            *(p+i+1) = temp;
            flag=1;
        }
    }
    k--;
}
} //chiusura sort

```

```

int readInt(char messaggio[])
{
    /*
    Autore : Riontino Raffaele 1/11/2010
    Funzionalità :
    funzione che ritorna un intero :
    questa funzione permette di inserire un intero,
    effettua un controllo sui dati inseriti, sul segno
    e sul valore.i dati vengono controllati carattere
    per carattere.
    */
    int massimo = 2147483646; //massimo intero
    int minimo = -2147483647; //minimo intero
    int dato = 0,i;
    bool errore;
    const int dim = 11;
    char stringa[dim],carattere;
    cout << messaggio; //messaggio inviato dall'utente
    do{
        errore = false; //inizializzo errore a false
        i = 0;
        do{
            carattere = getch(); //catturo un carattere
            if (carattere >= '0' && carattere <= '9' && i <= dim || (carattere
== '-' && i == 0))
            {
                cout << carattere;
                stringa[i] = carattere;
                i++;
            }
            if ((int)carattere == 8 && i > 0) //se tasto delete
            {
                cout << carattere << ' ' << carattere;
                i--;
            }
        }while ((int)carattere != 13); //ripeti se carattere diverso da invio
        stringa[i] = '\0'; //inserisco finestringa
        if ((atoi(stringa) >= minimo && atoi(stringa) <= massimo) && i < dim)
        dato = atoi(stringa);
        else
        {
            cout << "\n\a Numero non consentito. Ripeti!!!";
            errore = true;
        }
        if (!strlen(stringa)) cout << 0;
    }
}

```

```
    cout << endl;
  }while(errore);
  return dato;
} //chiusura readInt
```