**I.I.S. “B. Pascal” Reggio Emilia**

**VERIFICA DI INFORMATICA**

**Libreria Standard STL**

**Classe \_\_\_\_\_\_\_\_\_\_\_\_\_Nome\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data: 24 Ottobre 2018**

1. Dati i seguenti metodi per la classe vector illustrane il significato (p..……/10)

**size() \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**push\_back(**) **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**pop\_back()** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**begin()** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**clear() \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Data la seguente lista:

list<char> l;

e supponendo che contenga i seguenti elementi: a c b d h g i

scrivere nella colonna di destra il contenuto della lista l dopo l’esecuzione del codice indicato nella colonna a sinistra: (p………/9)

|  |  |
| --- | --- |
| CODICE | CONTENUTO DI l |
| list<char>::iterator i ;  l.push\_back(e);  for (i = l.begin(); i != l.end(); iter++)  cout << \*i <<endl; |  |
| list<char>::iterator i ;  sort( l.begin(),l.end() );  for (i = l.begin(); i != l.end(); iter++)  cout << \*i <<endl; |  |
| list<char>::iterator i ;  reverse( l.begin(),l.end() );  for (i = l.begin(); i != l.end(); iter++)  cout << \*i <<endl; |  |

1. Che cosa viene restituito dal metodo find() del contenitore map **m: m.find(T1)** ? (p……………/2)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Completa la seguente tabella specificando con una lettera A o con una lettera M se le funzioni elencate sono algoritmi (A) o metodi (M) per i contenitori indicati sulle colonne (p……./9)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **vector** | **list** | **map** |
| **count()** |  |  |  |
| **for\_each** |  |  |  |
| **unique()** |  |  |  |

1. Dato un vettore di numeri di dimensione terminante conla richiesta all’utente “Altro numero (0=FINE)?”,?”, memorizzare per ciascun numero in un altro vettore il doppio del numero se il numero è pari, il numero stesso se è dispari. Visualizzare il secondo vettore. Utilizzare la libreria STL. (p………/50)
2. Acquisire da tastiera un elenco di parole, memorizzandole in una lista, finchè l’utente segnala la fine dell’inserimento con l’”\*”.

Dichiarare le librerie necessarie per l’utilizzo della struttura lista, dichiarare la lista e svolgere SOLO inserimento in testa delle parole nelle due modalità. (p……./24)

|  |  |
| --- | --- |
| RISOLUZIONE SENZA LIBRERIA STL | RISOLUZIONE CON LIBRERIA STL |
| //librerie  //dichiarazione lista  // funzione inserimento in testa | //librerie  //dichiarazione lista  //funzione inserimento in testa |

1. Che cosa scrive in output il seguente programma? (p………/6)

//ese7.cpp

#include <iostream>

#include <map>

#include <string>

#include <iterator>

using namespace std;

int main()

{

map<string, int> mapOfWords;

mapOfWords.insert(make\_pair("earth", 1));

mapOfWords.insert(make\_pair("moon", 2));

mapOfWords["sun"] = 3;

mapOfWords["earth"] = 4;

map<std::string, int>::iterator it = mapOfWords.begin();

while(it != mapOfWords.end())

{

cout<<it->first<<" :: "<<it->second<<endl;

it++;

}

if(mapOfWords.insert(make\_pair("earth", 1)).second == false)

{

cout<<"Element with key 'earth' not inserted because already existed"<<endl;

}

if(mapOfWords.find("sun") != mapOfWords.end())

cout<<"word 'sun' found"<<endl;

if(mapOfWords.find("mars") == mapOfWords.end())

cout<<"word 'mars' not found"<<endl;

return 0;

}

OUTPUT:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_