

Problem Solving using c and c++

Lab Assignment 4

Patil Amit Gurusidhappa

19104004

B11

1. Write a C++ program to demonstrate different containers like vector, set, map, multi map etc. Use sort method to sort vectors in ascending and descending order.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    // vector
    vector<int> v = {3, 2, 5, 1};
    sort(v.begin(), v.end());
    for (int i = 0; i < (int)v.size(); i++)
    {
        cout << v[i] << " ";
    }
    cout << endl;
    // set
    set<int> s = {2, 3, 1, 2, 5};

    for (auto element : s)
    {
        cout << element << " ";
    }
    cout << endl;
    // map
    map<int, string> m = {

        {2, "Two"},
        {3, "Three"},
        {1, "one"},
    };

    for (auto element : m)
    {
```

```

        cout << element.first << " " << element.second << endl;
    }
    cout << endl;

    // multi map
    multimap<int, string> mm;
    mm.insert({1, "hello"});
    mm.insert({1, "hi"});

    for (auto element : mm)
    {
        cout << element.first << " " << element.second << endl;
    }
    cout << endl;
}

```

```

PS E:\Work\JIIT\sem_6\JIIT-SEM-6\Problem_solving_lab\Lab4> cd "e:\Work\
f ($?) { g++ q1.cpp -o q1 } ; if ($?) { .\q1 }
1 2 3 5
1 2 3 5
1 one
2 Two
3 Three

1 hello
1 hi

```

2. Write a C++ program to demonstrate working of `regex_match()`, `regex_search()`, and `regex_replace()`.

help: <https://www.softwaretestinghelp.com/regex-in-cpp/>

```

#include <bits/stdc++.h>
using namespace std;

int main()
{
    // Regex Match
    if (regex_match("softwareTesting", regex("(soft)(.*)")))
    {
        cout << "matched" << endl;
    }
}

```

```

else
{
    cout << "Not Matched" << endl;
}

/* regex search
// string to be searched
string mystr = "She sells_sea shells in the sea shore";

// regex expression for pattern to be searched
regex regexp("s[a-z_]+");

// flag type for determining the matching behavior (in this case on
string objects)
smatch m;

// regex_search that searches pattern regexp in the string mystr
regex_search(mystr, m, regexp);

cout << "String that matches the pattern:" << endl;
for (auto x : m)
    cout << x << " ";

/* regex match
string mystr1 = "This is software testing Help portal \n";

cout << "Input string: " << mystr1 << endl;

// regex to match string beginning with 'p'
regex regexp1("p[a-zA-z]+");
cout << "Replace the word 'portal' with word 'website' : ";
// regex_replace() for replacing the match with the word 'website'
cout << regex_replace(mystr1, regexp1, "website");

string result;

cout << "Replace the word 'website' back to 'portal': ";
// regex_replace( ) for replacing the match back with 'portal'
regex_replace(back_inserter(result), mystr1.begin(), mystr1.end(),
              regexp1, "portal");

```

```

    cout << result;
}

```

```

PS E:\Work\JIIT\sem_6\JIIT-SEM-6\Problem_solving_lab\Lab4> cd "e:\Work\JIIT\sem_6\JIIT-SEM-6\Problem_solving_lab\Lab4"
f ($?) { g++ q2.cpp -o q2 } ; if ($?) { .\q2 }
matched
String that matches the pattern:
sells_sea Input string: This is software testing Help portal

Replace the word 'portal' with word 'website' : This is software testing Help website
Replace the word 'website' back to 'portal': This is software testing Help portal

```

3. Write a C++ program to

- i. Match an underscore (`_`) or an uppercase or lowercase letter.
- ii. Match any number of underscore, letter, or digit characters.
- iii. Match a literal dot (`.`).
- iv. Again, match any number of underscore, letter, or digit characters.

```

#include <iostream>
#include <string>
#include <regex>
using namespace std;

int main()
{
    if (regex_match("softwareTesting", regex("(soft)(.*)")))
        cout << "string:literal => matched\n";

    const char mystr[] = "SoftwareTestingHelp";
    string str("software");
    regex str_expr("(soft)(.*)");

    if (regex_match(str, str_expr))
        cout << "string:object => matched\n";

    if (regex_match(str.begin(), str.end(), str_expr))
        cout << "string:range(begin-end) => matched\n";

    cmatch cm;
    regex_match(mystr, cm, str_expr);

    smatch sm;
}

```

```

    regex_match(str, sm, str_expr);

    regex_match(str.cbegin(), str.cend(), sm, str_expr);
    cout << "String:range, size:" << sm.size() << " matches\n";

    regex_match(mystr, cm, str_expr, regex_constants::match_default);

    cout << "the matches are: ";
    for (unsigned i = 0; i < sm.size(); ++i)
    {
        cout << "[" << sm[i] << "]" ";
    }

    cout << endl;

    return 0;
}

```

Output

```

PS E:\Work\JIIT\sem_6\JIIT-SEM-6\Problem_solving_lab\Lab4> cd "e:\Work\JIIT\sem_6\JIIT-SEM-6\Problem_solving_lab\L
ab4\" ; if ($?) { g++ q3.cpp -o q3 } ; if ($?) { .\q3 }
string:literal => matched
string:object => matched
string:range(begin-end)=> matched
String:range, size:3 matches
the matches are: [software] [soft] [ware]

```