POPL Assignment: 4

Aryan Agarwal, 19CS30005

(a) (i) A technique in C++ that allows us to write inline functions without a name.

(b) (i) The return type of lambda expressions can be neglected in some cases.

(c) (iii) 3

(d)
```
# include <iostream>
# include <functional>
# include <algorithm>
# include <vector>
using namespace std;
int main (){
  vector <int> v = {7,5,3,6,-10,10}
  Sort (v.begin(), v.end(), [] (const int &a, const int &b)
    {
      return a>b;
    }
  );
  for (const auto &x : v) cout <<x<<endl;
  return 0;
}
```

(e)
```
# include <iostream>
# include <functional>
# include <algorithm>
# include <vector>

using namespace std;

int main() {

    vector <int> v = {1,2,3,4,5,6,7,8};

    int count = count_if (v.begin(), v.end(),
                [] (int x) {return x%2==0;});

    cout << count << endl;
    return 0;
}
```

(f)
```
# include <iostream>
# include <functional>
using namespace std;
int main() {
    int x=0
    auto make_one = [&].() { x=1; };

    make_one();

    cout << x << endl;
    return 0;
}
```

2.(a)

```
#include <iostream>
using namespace std;
class TowerOfHanoi{
public:
    void operator() (int n, char f, char aux, char t){
        if(n<0) return;
        if(n==1){
            cout<<"Move disc1 from"<<f<<"to"<<t<<endl;
            return;
        }
        (*this)(n-1, f, t, aux);
        cout<<"Move disc"<<n<<"from"<<f<<"to"<<t<<endl;
        (*this)(n-1, aux, f, t);
    }
};
int main(){
    TowerOfHanoi t;
    int n;
    cout<<"Enter no of discs :";
    cin>>n;
    t(n, 'A', 'B', 'C'); //Rods are name A, B and C
    return 0;
}
```

2.(b)
```
#include <iostream>
#include <functional>
using namespace std;
int main(){
std::function<void(int,char,char,char)>TowerOfHanoi;
TowerOfHanoi=[&TowerOfHanoi](int n, char f, char aux, char t){
    if (n<0) return;
    if(n==1){
      cout<<"Move disc1 from"<<f<<"to"<<t<<endl;
      return;
    }
    TowerOfHanoi(n-1, f, t, aux);
    cout<<"Move disc"<<n<<"from"<<f<<"to"<<t<<endl;
    TowerOfHanoi(n-1, aux, f, t);
};
int n;
cout<<"Enter no. of discs";
cin>>n;
TowerOfHanoi(n, 'A','B','C'); //Rods are A, B, C
return 0;
}
```

(a)
```
#include <iostream>
#include <string>
using namespace std;

class Permute {
public:
    void operator() (string s, int l, int r) {
        if (l == r) {
            cout << s << endl;
            return;
        }
        for (int i = l; i <= r; i++) {
            swap(s[l], s[i]);
            (*this)(s, l+1, r);
            swap(s[l], s[i]);
        }
    }
};

int main() {
    cout << "Enter string: ";
    string s;
    cin >> s;
    int n = (int) s.size();

    Permute permute;
    permute(str, 0, n-1);

    return 0;
}
```

(b)
```
#include <iostream>
#include <string>
using namespace std;

int main(){
std::function<void(string,int,int)> permute;
permute =[&permute](string s, int l, int r){
        if (l==r){
            cout<<s<<endl;
        }   return;

    for(int i=l; i<=r; i++){
      swap(s[l],s[i]);
      permute(s, l+1, r);
      swap[s[l],s[i]);
    }

};
cout<<"Enter string";
    string s;
    cin>>s;
int n= (int) s.size();
permute(s,0,n-1);
return 0;
}
```