1.

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

void palindrome(long int n) {
  int rev=0,temp;
  temp = n;
  while(n > 0) {
     rev = rev * 10 + n % 10;
     n = n / 10;
  }
  if(temp==rev)
  cout<<temp<<" is a palindrome"<<endl;
  else
  cout<<temp<<" is not a palindrome"<<endl;
}
int main() {
  long int n;
  cout<<"Enter the string : ";
  cin>>n;
  palindrome(n);
  return 0;
}
```

2.

```
#include <iostream>
#include <string>
#include <cstring>
#include <sstream>
using namespace std;

int main() {
  string IP_1;
 cin>>IP_1;
  string IP_2;
 cin>>IP_2;
 string r1 = IP_1.substr(10,3);
 string r2 = IP_1.substr(14,2);
 string  r3 = IP_2.substr(10,4);
  stringstream range_1(r1);
 stringstream range_2(r2);
 stringstream range_3(r3);
 int range1 = 0;
 int range2 = 0;
 int range3 = 0;
 range_1>> range1;
 range_2>> range2;
 range_3>> range3;

 string cmp1 = IP_1.substr(0,9);
 string cmp2 = IP_2.substr(0,9);
  if(cmp1 == cmp2){
   if( range2 <= range3 && range1 >= range3){
     cout<<"IP Matched";
   }
   else{
     cout<<"Not Matched";
   }
 }
  else{
   cout<<"Not Matched";
 }
  return 0;
}
```

3.

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

int main() {
   float dur = 0;
   float rem = 0;
   float cost = 0;
   float inc = 0;
   float inc_cost = 0;
   float rate = 0.05;

   cout<<"Enter Duration : "<<endl;
   cin>>dur;

   rem = dur - 60;

   if(dur==0 && dur<0){
       cost = rate;
   }

   else{
       inc = round(rem/30);
       inc_cost = rate*inc;
       cost = rate + rate*inc;
   }

   cout<<"Cost of Incremental Duration : "<< inc_cost <<"$"<<endl;
   cout<<"Cost of Connection Duration : "<<cost<<"$"<<endl;

   return 0;
}
```

4.

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

int main() {
   float calls = 0;
   int server;
   float server_1 = 0;
   float server_2 = 0;
   float server_3 = 0;
   float server_4 = 0;

   cout<<"Enter Total Calls"<<endl;
   cin>>calls;
   float temp = calls;

   if(calls == 0){
       cout<<"All servers are full";
   }

   else{
       for(int i=0;i<calls;i++){

       cin>>server;

       switch(server){
       case 1:
       server_1++;
       temp--;
       break;

       case 2:
       server_2++;
       temp--;
       break;

       case 3:
       server_3++;
       temp--;
       break;
```

```
        case 4:
        server_4++;
        temp--;
        break;

        default:
        cout<<"Wrong Input"<<endl;
        break;

        }
    }
    }

    cout<<"Server 1 : "<<server_1<<endl;
    cout<<"Server 2 : "<<server_2<<endl;
    cout<<"Server 3 : "<<server_3<<endl;
    cout<<"Server 4 : "<<server_4<<endl;

    return 0;
}
```

5.

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

int top;
char status[3][6];
int vacant[18];

   int CAR_IN(){
       if(top==-1){
           cout<<"No Space"<<endl;
       }
       else{
           int row = (vacant[top]/6);
           int col = vacant[top]%6;
           status[row][col] = '*';
           vacant[top] = -1;
           top = top-1;
       }

       return 0;

   }

   int CAR_OUT(int position){
       int row = position/6;
       int col = position%6;
       if(status[row][col] == '0'){
           cout<<"Already Vacant"<<endl;
       }
       else{
           status[row][col] = '0';
           top = top+1;
           vacant[top] = position;
       }
       return 0;
   }

   int CAR_STATUS(){
```

```
    for(int i=0; i<3; i++){
        for(int j=0; j<6; j++){
            cout<<status[i][j]<<" ";
        }
        cout<<endl;
    }

    return 0;

    }


int main() {
    int position;
    top = 17;
    for(int i=0; i<18; i++){
        vacant[i]= i;
    }

    for(int i=0; i<3; i++){
        for(int j=0;j<6;j++){
            status[i][j] = '0';
        }
    }

    int choice;

    cout<<"Press 1 for CAR IN"<<endl;
    cout<<"Press 2 for CAR OUT"<<endl;
    cout<<"Press 3 for CAR STATUS"<<endl;

    CAR_IN();
    CAR_IN();
    CAR_IN();
    CAR_IN();
    CAR_IN();

    cin>>choice;

    if(choice == 1){
```

```
        CAR_IN();
        CAR_STATUS();
    }

    else if(choice == 2){
        cin>>position;
        CAR_OUT(position);
        CAR_STATUS();
    }

    else if(choice == 3){
        CAR_STATUS();
    }


    return 0;
}
```

6.

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


int main()
{
   string line;
   const char* WhiteSpaces = " \t\v\r\n";

   getline(cin, line);

   size_t start = line.find_first_not_of(WhiteSpaces);
   size_t end = line.find_last_not_of(WhiteSpaces);

   string answer = start == end ? string() : line.substr(start, end -
start + 1);
   cout<< answer <<endl;
}
```

Linux Commands

```
1. ls -a

2. :set nu

3. df

4. uname

5. chown
```

OOPS concept

1. Basic concepts of OOPS are :
a). Classes
b). Objects
c). Encapsulation
d). Polymorphism
e). Inheritance
f). Abstraction

2. Friend Function : Friend function is a function which is declared outside of the class but it can access all the private and protected members of class.

3. Difference :
Class - A class is a template of creating objects in program.
Object - Object is an instance of the class.

4. Overriding : If a child class is using the same method as declared in the parent class, it is knows as Overriding.
Example -

```
class Human {
   public:
   void walk(){
       cout<<"walking";
   }
};

class Male: public Human{
   public:
   void walk(){
       cout<<"Male walking";
   }
};
```

5. This Pointer - This pointer points and allows the object to access its address.

Quantitative Test

```
1. C. 37

2. C. 427

3. B. 193

4. C. 12

5. D. None of These
```