

Ex. 6.2d

```

#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    cout << "Received " << argc << " arguments." << endl;
    for (int i = 0; i<argc; i++)
        cout << "argument " << i << ": " << argv[i] << endl;
    return 0;
}
/*   Ex6p2d aa bb param3 param4
    The output will show:

```

```

Received 5 arguments.
argument 0:   Ex6p2d
argument 1:   aa
argument 2:   bb
argument 3:   param3
argument 4:   param4

```

```

    */

```

Ex. 6.2e

```

#include <iostream>
#include <string.h>
using namespace std;
int main(int argc, char *argv[])
{
    bool sameoption=false;
    cout << "Received " << argc << " arguments." << endl;
    for (int i = 0; i<argc; i++)
    {
        cout << "argument " << i << ": " << argv[i] << endl;
        for (int j=i+1; j<argc; j++)
            if (strcmp (argv[i], argv[j])==0)
            {
                sameoption=true; break; // Same option detected
            }
    }
    if (sameoption == true)
        cout << "Warning: Same command-line options detected!"<<endl;
    return 0;
}

```

Ex. 6.3

```
#include <iostream>
#include <string.h>
using namespace std;
class CAT
{
    public:
        CAT(char * firstname) {strncpy(name,firstname,79);}
        ~CAT() {}
        char * GetName() {return name;}
        void SetName(char *nameinput) {strncpy(name,nameinput,79);}
    private:
        char name[80];
};

void nameswap(CAT *CatA, CAT *CatB);

int main()
{
    CAT Cat1("Frisky"), Cat2("Felix");
    cout << "Before swap, Cat1: " << Cat1.GetName() << "\tCat2: " <<
Cat2.GetName() << "\n";
    nameswap(&Cat1,&Cat2);
    cout << "After swap, Cat1: " << Cat1.GetName() << " \tCat2: " <<
Cat2.GetName() << "\n";
    return 0;
}

void nameswap(CAT *CatA, CAT *CatB)
{
    char temp[80];
    strncpy(temp, CatA->GetName(),79);
    CatA->SetName(CatB->GetName());
    CatB->SetName(temp);
}
```