

1. (2 pt.)

```

all: abcd

abcd: a.o b.o c.o d.o
      g++ a.o b.o c.o d.o -o abcd

a.o: a.cpp
     g++ -c a.cpp

b.o: b.cpp
     g++ -c b.cpp

c.o: c.cpp
     g++ -c c.cpp

d.o: d.cpp
     g++ -c d.cpp

clean:
      rm -rf *.o abcd
    
```

2. (3 pt.) What are outputs from each of following simple program segment?

```

//a)
#include <iostream>
using namespace std;
int main()
{
    int A[6]={2, 4, 6, 8, 10};
    int *ptr = A+2;
    cout << *(ptr+2)<<endl;
    return 0;
}
    
```

10

```

//b)
#include <iostream>
using namespace std;
int main()
{
    int A[6]={2, 4, 6, 8, 10};
    cout << A+3 <<endl;
    return 0;
}
    
```

Address of 8

```
//c)
#include <iostream>
using namespace std;
int main()
{
    char A[]="GOOD MORNING";
    char *ptr =A;
    cout << ptr+3<<endl;
    return 0;
}
```

D MORNING

```
//d)
#include <iostream>
using namespace std;
int main()
{
    char A[8]="GOOD MORNING";
    char *ptr =A+7;
    cout << *ptr<<endl;
    return 0;
}
```

R

```
//e)
#include <iostream>
using namespace std;
int main()
{
    char *ptr[4]={"YOU", "ARE", "MY", "LOVE"};
    cout << ptr[1]+1 <<endl;
    return 0;
}
```

RE

```
//f)
#include <iostream>
using namespace std;
int main()
{
    char *ptr[4]={"YOU", "ARE", "MY", "LOVE"};
    cout << *(ptr[3]+2)<<endl;
    return 0;
}
```

V

3. (2 pt.)

```
bool cstrEq (const char A[], const char B[])
{
    int i=0;
    bool rval=false;
    while ((A[i]==B[i]) && (A[i]!='\0') && (B[i]!= '\0'))
        i++;
    if (A[i]== B[i])
        rval=true;

    return rval;
}
```

4. (3 pt.).

```
Node *insertNode (Node *list)
{
    Node *tmp = list; // temporary pointer to traverse linked list
    Node *newNode = new Node;
    cout << "a integer value: " << endl;
    cin >> newNode->info;
    // list is empty
    if (list == NULL)
        list = newNode;

    else
    {
        //Find the last node or duplication
        while ((tmp->next != NULL) && ((tmp->info != newNode->info))
            tmp = tmp->next;
        if (tmp->info == newNode->info)
        {
            cout << newNode->info << " is already in the list! " << endl;
            delete newNode;
        }
        else
            //insert as the last element
            tmp->next = newNode;
    }
    return list;
}
```