1. (10 pt.)

   a) f(n) = O (g(n))
      - O(g(n))= {f (n) | there exist positive constant c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n0$}. g(n) is upper bound of f(n)

   b) f(n) = Θ (g(n))
      - Θ(g(n)) = {f (n) | there exist positive constant c_1, c_2 and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$}. g(n) is tight bound of f(n)

   c) f(n) = Ω (g(n))

      - Ω (g(n)) = {f (n) | there exist positive constant c and n_0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0$}. g(n) is lower bound of f(n)

2. (5 pt.)

   Sol) By asymptotic notation, we need show that there exist constant *C_1, C_2* and *n_0* such that

$$C_1n^2 \leq n^2 + 2n \leq C_2n^2 \;\; for\ all\;\; n_0 \geq 0$$
$$C_1n^2 \leq n^2 + 2n, \qquad where\ C_1 = 1\ with\ n_0 = 1$$
$$n^2 + 2n \leq C_2n^2, where\ C_2 = 2\ with\ n_0 = 2$$
$$C_1n^2 \leq n^2 + 2n \leq C_2n^2\ with\ C_1 = 1, C_2 = 2, n_0 = 2\ true$$

3. (5 pt.)

   Sol) By asymptotic notation, there exist constant *C,* and *n_0* such that

$$\frac{1}{2}n^2 - 3n \leq Cn^2 \;\; with\ C = 1\ and\ n_0 = 1, it\ is\ true$$

4. (15 pt.) What will each of the following programs display on the screen?

a) Different

```
#include <iostream>
using namespace std;
int main()
{
      char yourName[20] = "Mary";
      char myName[20]="Mary";
      if (yourName == myName)
            cout << "Same"<<endl;
      else
            cout <<"Different"<<endl;
      return 0;
}
```

b)

```
#include <iostream>
using namespace std;
int main()
{
   int num1 = 5;
   int num2 = 5;
   cout <<--num1 <<++num2<<endl;
   cout <<num1-- << num2++<<endl;
   return 0;
}
```

4 6
4 6

c)

```
#include <iostream>
using namespace std;
int main ()
{
   int cnt;
   for (cnt = 1; cnt <= 10; cnt++){
       if (!(cnt % 4 == 0))
            cout << cnt << " "<<endl;
    }
   retun 0;
}
1 2 3 5 6 7 9 10
```

d)

```
#include <iostream>
using namespace std;
int main()
{
      char *ptr ={"good for you?"};

      cout << ptr+1 <<endl;
      cout << *(ptr+2) <<endl;
      return 0;
}
ood for you
o
```

e)

```
#include <iostream>
using namespace std;
int main ()
{
   for (int i = 1; i <= 10; i++){
        if (i % 4 ==0)
           break;
        cout << i << ' ';
  }
  return 0;
}   1 2 3
```

f)

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

int main()
{
     char *ptr[2] ={"123456", "789" };
     cout << ptr[0]+2 <<endl;
     return 0;

}
```

3456

g)

```
#include <iostream>
using namespace std;
int main()
{
      int a[5] ={1,2,3,4,5};
      int *ptr;
      ptr = a+1;
      ptr=ptr+2;
      cout << *ptr<<endl;
      return 0;
}   4
```

h)

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

int main()
{
   char *ptr[2] ={"ABCDEF", "GHIJK" };
   cout << *(ptr+1) <<endl;
   return 0;

}
```

GHIJK

5. (15 pt.)

```
bool Palindrome (char A[], int Size)
{
    int front =0;
    int back =Size -1;
    bool rval;
    while ((A[front]==A[back]) && (front <= back))
    {
        front++;
        back --;
    }
    if (front > back)
        rval = true;
    else
        rval = false;
    return rval;


}
```

6. (10 pt)

Insertion Sort	Selection Sort	Bubble Sort
8, 6, 2, 3, 4, 5, 7	8, 6, 2, 3, 4, 5, 7	8, 6, 2, 3, 4, 5, 7
6, 8, 2, 3, 4, 5, 7	2, 6, 8, 3, 4, 5, 7	6, 2, 3, 4, 5, 7, 8
2, 6, 8, 3, 4, 5, 7	2, 3, 8, 6, 4, 5, 7	2, 3, 4, 5, 6, 7, 8
2, 3, 6, 8, 4, 5, 7	2, 3, 4, 6, 8, 5, 7	2, 3, 4, 5, 6, 7, 8
2, 3, 4, 6, 8, 5, 7	2, 3, 4, 5, 8, 6, 7	2, 3, 4, 5, 6, 7, 8
2, 3, 4, 5, 6, 8, 7	2, 3, 4, 5, 6, 8, 7	2, 3, 4, 5, 6, 7, 8
2, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 6, 7, 8

7. (10 pt.)

.

```
void selectionSort (int A[], int size)
{
     int min,tmp;
     for (int j = 0; j < size -1; j++)
     {
          min = j;
          for (int k= j+1; k <= size -1; k++)
          {
               if (A [k]  < A[min])
                    min = k;
          }
          if (min != j)
          {
               tmp = A[j];
               A[j] = A[min];
               A[min] = tmp;
          }
     }
}
```

8. (5 pt.)

   (Answer)

   - Text section – executable code
   - Data section – Global variable and Static variables
   - Heap – space for dynamic memory allocation
   - Stack section – store local variables when a function all

9. (15 pt)

```
Node *DeleteNode (Node *List, int ID)
{
      Node *Tmp = List;

      // Case 1: List is empty
      if (List == NULL)
            cout << "Empty List" <<endl;
      //Case 2: Delete the first node
      else if (Tmp->IDNumber == ID)
      {
            List = List->Next;
            delete Tmp;
      }
      else
      {
            while (Tmp->Next !=NULL && Tmp->Next->ID != ID)
                  Tmp = Tmp->Next;
            // Case 3: No Such a node
            if (Tmp ->Next == NULL)
                  cout <<"There is no student with ID number: "<<ID <<endl;
            // Case 4: Delete a node from a list (not first node)
            else if (Tmp ->Next->IDNumber == ID)
            {
                  Node *T = Tmp->Next;
                  Tmp->Next = Tmp->Next->Next;
                  delete T;
            }
      }
      return List;
}
```

10. (10 pt.)

```
#include <iostream>
using namespace std;
void funA(int, int);
void funB(int);

class test {
public:
   int a;
   test(int);
   ~test();
};
// constructor
test::test(int x)
{
   a = x;
   cout << a <<" is created" <<endl;
}
// destructor
test::~test()
{
   cout << a << " is destroyed" <<endl;
}
```

```
// main
int main()
{
    test a(10);
    funA(20, 30);
    funB(40);
    return 0;
}
//funA
void funA(int x, int y)
{
    test a(x);
    test *ptr = new test (x+1);
    funB (y);
    delete (ptr);

}

// funB
void funB(int x)
{
   test a(x);
   test *ptr = new test(x+1);
}
```

10 is created
20 is created
21 is created
30 is created
31 is created
30 is destroyed
21 is destroyed
40 is created
41 is created
40 is destroyed
10 is destroyed