```
C++ Ex1.cpp
           ×
               C++ Ex2.cpp
                                 C++ Ex3.cpp
C** Ex1.cpp > 分 main()
        #include <iostream>
        using namespace std;
   2
        int NDsearch(int A[], int n, int key)
   4
            int i;
   5
   6
            for (i = 0; i < n; i++)
   7
                if (key == A[i])
   8
   9
                     break;
  10
  11
  12
            if (i < n)
  13
  14
                cout << "found";</pre>
  15
  16
            else
  17
  18
                cout << "there is no element";</pre>
  19
  20
  21
        int main()
  22
  23
            int A[] = \{12, 34, 56, 74, 11, 13, 10, 20\};
  24
            int key = 11;
  25
            NDsearch(A, 8, key);
  26
  27
```

iğ uhu du lun villi mişu müşdlə	mid	មុខវិទូរ
Exercise 1. Non- Questicate		
(Start)		
/ALI, Key		
/NDSearch(A, n, key)/		
j (n)		/P
Key ≠ ¥ADD →	1 RN	→ / p 8 (1(\)
	rs no (le ment/
1-11+		

```
C++ Ex2.cpp
C++ Ex1.cpp
                                 C++ Ex3.cpp
C++ Ex2.cpp > 🗇 Dsearch(int [], int, int)
   4
   5
            int i;
   6
            int count = 0;
   7
            for (i = 0; i < n; i++)
   8
                if (key == A[i])
   9
  10
  11
                     count++;
  12
  13
  14
            if (count > 0)
  15
                cout << "found there are " << count << " element in array";</pre>
  16
  17
            else
  18
  19
  20
                cout << "there is no element";</pre>
  21
  22
        int main()
  23
  24
  25
            int A[] = {12, 34, 56, 74, 11, 13, 10, 20, 20, 20, 20};
  26
            int key = 20;
            Dsearch(A, 11, key);
  27
  28
```

i dig n	ក្សា ប្រហ សុក្រ សៅ អាច្បៃ កាលជំ	រ៉េង	1013	មុខវិជ្ជា
Exerci.	se 2. Puplicate			
	(Start)			
//	A carray, key=20	7		
	DSearch (A, n, k	ey)/		
	/i=0, count=0/			
	izn , w	(n+)0)_	→ I fou	ind, point
	Xes	(elemen	v+/6r
ķ	Key=A[i] Xes/Coun.	++/		
j		1		1
11	Mm			
	/no			
9	/1++/<			
	/1++/<			
	71+-7<			
	/1++/<			
	/1++/<			

```
C++ Ex2.cpp • C++ Ex3.cpp
C++ Ex1.cpp
C→ Ex3.cpp > 分 IBsearch(int [], int, int, int)
       #include <iostream>
       using namespace std;
       int RBsearch(int A[], int l, int h, int key)
            int curin = (1 + h) / 2;
            if (A[curin] < key)</pre>
                return RBsearch(A, curin + 1, h, key);
   8
   9
  10
            else if (A[curin] > key)
  11
                return RBsearch(A, l, curin - 1, key);
  12
  14
            return curin;
  15
       int IBsearch(int A[], int l, int h, int key)
  16
  17
  18
            int curin;
            while (1 < h)
  19
  20
  21
                curin = (1 + h) / 2;
                if (A[curin] > key)
  22
  23
                    h = curin - 1;
  24
  25
                else if (A[curin] < key)</pre>
  26
  27
                  l = curin + 1;
  28
  30
                else
  31
                    return curin;
  34
  35
  36
  37
       int main()
  38
  39
            int A[] = \{1, 2, 3, 4, 5, 6, 7, 8, 9\};
            int key = 9; // key search
  40
            cout << "the element found at index " << RBsearch(A, 0, 8, key) << " of array" << endl;
  41
            cout << "the element found at index " << IBsearch(A, 0, 8, key) << " of array";</pre>
  42
  43
```

