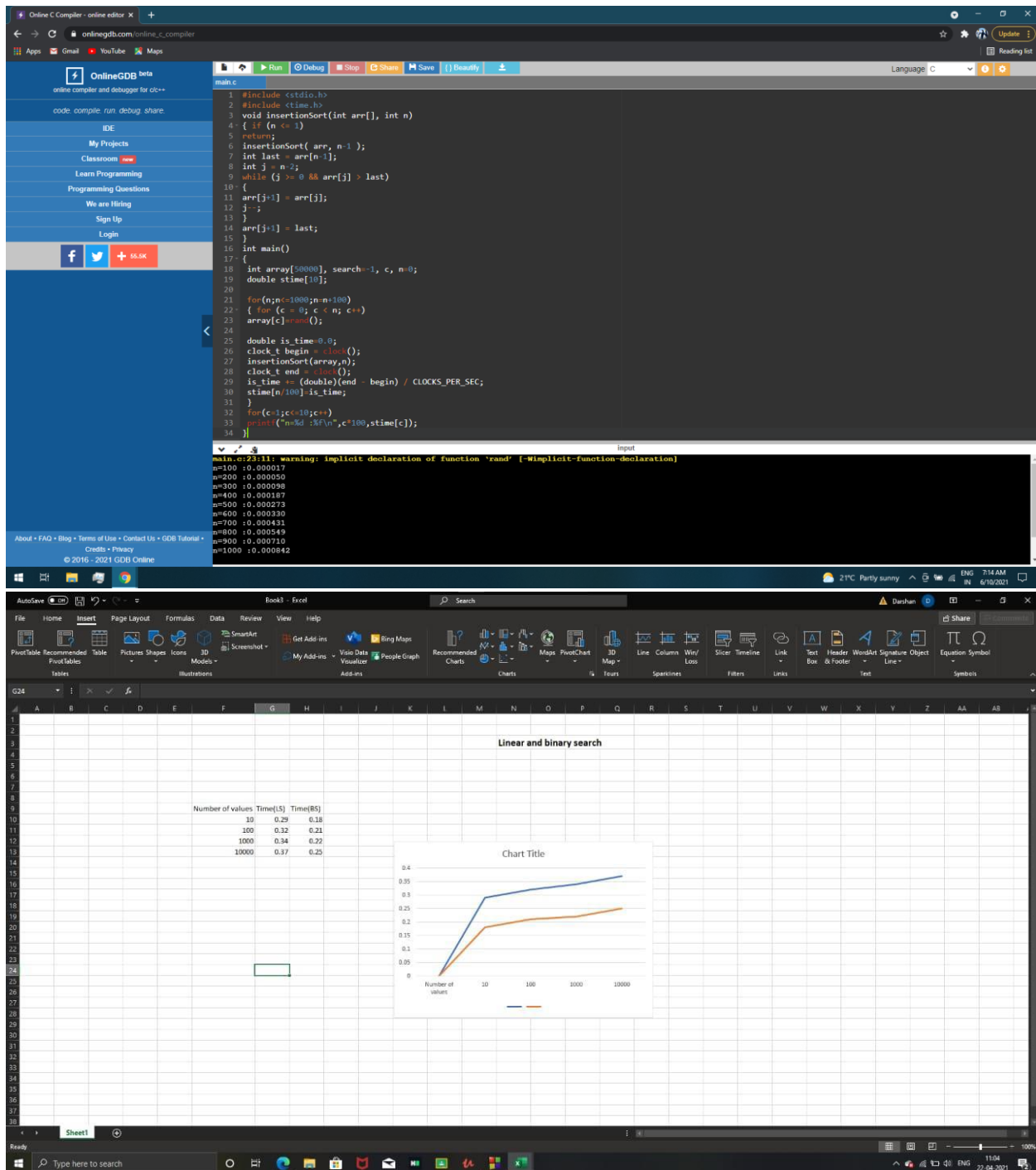


## OUTPUT FOR INSERT SORT



## CODE for insert sort

```
#include <stdio.h>
```

```
#include <time.h>
```

```
void insertionSort(int arr[], int n)
```

```

{ if (n <= 1)
return;

insertionSort( arr, n-1 );

int last = arr[n-1];

int j = n-2;

while (j >= 0 && arr[j] > last)
{
arr[j+1] = arr[j];

j--;
}

arr[j+1] = last;
}

int main()
{

int array[50000], search=-1, c, n=0;

double stime[10];


for(n;n<=1000;n=n+100)
{ for (c = 0; c < n; c++)
array[c]=rand();


double is_time=0.0;

clock_t begin = clock();

insertionSort(array,n);

clock_t end = clock();

is_time += (double)(end - begin) / CLOCKS_PER_SEC;

stime[n/100]=is_time;

}

for(c=1;c<=10;c++)

```

```
printf("n=%d :%f\n",c*100,stime[c]);
}
```

## OUTPUT FOR SEARCH

The image displays two screenshots of the OnlineGDB compiler interface, showing the execution of a C program for a search algorithm.

**Top Screenshot:** The code defines two search functions: `linear_search` and `binary_search`. The `main` function prompts the user for the size of the array (`n`), the element to search (`k`), and a choice between linear and binary search. The output shows the user entering `3` for the array size, `2` for the element, and choosing `2` for binary search. The result is "Element not found".

```
1 #include<stdio.h>
2 #include <time.h>
3 clock_t start, end;
4 int linear_search(int arr[],int n,int k)
5 {
6     if(n==0)
7         return 0;
8     if(arr[n] == k)
9         return 1;
10    else
11        return 0;
12    return linear_search(arr,n-1,k);
13 }
14 int binary_search(int arr[],int start,int end,int k)
15 {
16     if(start>end)
17         return 0;
18     int mid = (start + end )/ 2;
19     if(arr[mid]==k)
20         return 1;
21     else if(arr[mid]<k)
22         end = mid - 1;
23     else
24         start = mid + 1;
25     return binary_search(arr,start,end,k);
26 }
27 int main()
28 {
29     int n,i;
30     printf("Enter the size of array:");
31     scanf("%d",&n);
32     int choice;
33     int arr[n],k;
34     printf("\n");
```

**Bottom Screenshot:** The code is extended to include timing and a loop for multiple searches. It uses `clock_t` to measure the time taken for each search. The output shows the user entering `3` for the array size, `2` for the element, and choosing `2` for binary search. The result is "Element not found".

```
35     printf("Enter the element to search:");
36     scanf("%d",&k);
37     while(choice!=3)
38     {
39         printf("1 - Linear search\n");
40         printf("2 - Binary search\n");
41         printf("3 - Exit\n");
42         printf("Enter your choice:");
43         scanf("%d",&choice);
44         if(choice==1)
45         {
46             start=clock();
47             for(i=0;i<100000000;i++);
48             int a = linear_search(arr,n-1,k);
49             end=clock();
50             if(a==1)
51                 printf("Element found\n");
52             else
53                 printf("Element not found\n");
54             printf("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
55         }
56         else if(choice==2)
57         {
58             start=clock();
59             for(i=0;i<100000000;i++);
60             int b = binary_search(arr,0,n-1,k);
61             end=clock();
62             if(b==1)
63                 printf("Element found\n");
64             else
65                 printf("Element not found\n");
66             printf("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
67         }
68         else if(choice==3)
```

Online C Compiler - online editor

onlinegdb.com/online\_c\_compiler

OnlineGDB beta  
online compiler and debugger for C/C++

code: compile, run, debug, share.

IDE

My Projects

Classrooms

Learn Programming

Programming Questions

We are Hiring

Sign Up

Login

Facebook Twitter +55 SK

```
main.c
41 print("3 - Exit\n");
42 print("Enter your choice:");
43 scanf("%d",&choice);
44 if(choice==1)
45 {
46 start=clock();
47 for(i=0;i<800000000;i++);
48 int a = linear_search(arr,n-1,k);
49 end=clock();
50 if(a==1)
51 print("Element found\n");
52 else
53 print("Element not found\n");
54 print("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
55 }
56 else if(choice==2)
57 {
58 start=clock();
59 for(i=0;i<800000000;i++);
60 int b = binary_search(arr,0,n-1,k);
61 end=clock();
62 if(b==1)
63 print("Element found\n");
64 else
65 print("Element not found\n");
66 print("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
67 }
68 else if(choice==3)
69 break;
70 else
71 print("Invalid choice:");
72 }
73 }
74 }
```

Input

Enter the size of array:3

Enter the element to search:2

1 - Linear search  
2 - Binary search  
3 - Exit

Enter your choice:2

Element not found

Time taken to search value in 3 numbers is 0.170609 Secs

1 - Linear search  
2 - Binary search

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial •  
Credits • Privacy  
© 2016 - 2021 GDB Online

21°C Partly sunny 7:24 AM 6/10/2021

Online C Compiler - online editor

onlinegdb.com/online\_c\_compiler

OnlineGDB beta  
online compiler and debugger for C/C++

code: compile, run, debug, share.

IDE

My Projects

Classrooms

Learn Programming

Programming Questions

We are Hiring

Sign Up

Login

Facebook Twitter +55 SK

```
main.c
26 }
27 int main()
28 {
29 int n,i;
30 print("Enter the size of array:");
31 scanf("%d",&n);
32 int choice;
33 int arr[n];
34 print("\n");
35 print("Enter the element to search:");
36 scanf("%d",&k);
37 while(choice<=1)
38 {
39 print("1 - Linear search\n");
40 print("2 - Binary search\n");
41 print("3 - Exit\n");
42 print("Enter your choice:");
43 scanf("%d",&choice);
44 if(choice==1)
45 {
46 start=clock();
47 for(i=0;i<800000000;i++);
48 int a = linear_search(arr,n-1,k);
49 end=clock();
50 if(a==1)
51 print("Element found\n");
52 else
53 print("Element not found\n");
54 print("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
55 }
56 else if(choice==2)
57 {
58 start=clock();
59 for(i=0;i<800000000;i++);
60 int b = binary_search(arr,0,n-1,k);
61 end=clock();
62 if(b==1)
63 print("Element found\n");
64 else
65 print("Element not found\n");
66 print("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-start))/CLOCKS_PER_SEC));
67 }
68 else if(choice==3)
69 break;
70 else
71 print("Invalid choice:");
72 }
73 }
```

Input

Enter the size of array:3

Enter the element to search:2

1 - Linear search  
2 - Binary search  
3 - Exit

Enter your choice:2

Element not found

Time taken to search value in 3 numbers is 0.170609 Secs

1 - Linear search  
2 - Binary search  
3 - Exit

Enter your choice:1

Element not found

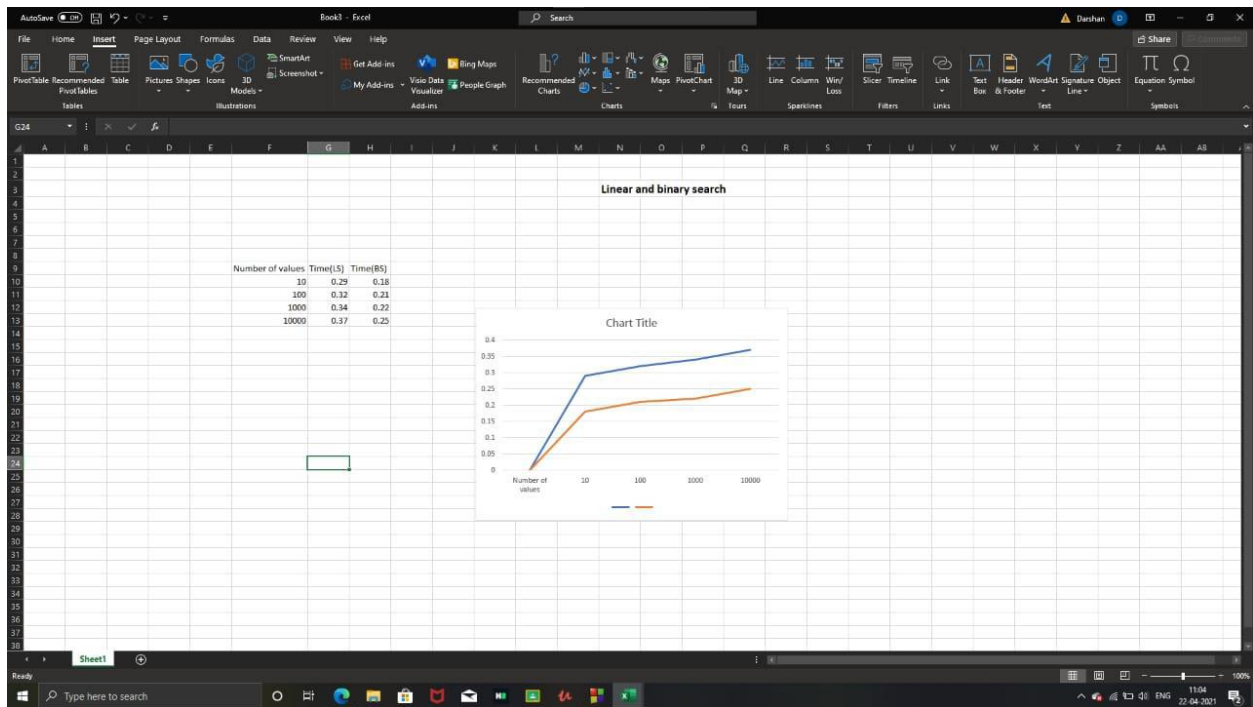
Time taken to search value in 3 numbers is 0.168888 Secs

1 - Linear search  
2 - Binary search  
3 - Exit

Enter your choice:

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial •  
Credits • Privacy  
© 2016 - 2021 GDB Online

21°C Partly sunny 7:26 AM 6/10/2021



## CODE FOR SEARCH

```
#include<stdio.h>

#include <time.h>

clock_t start, end;

int linear_search(int arr[],int n,int k)
{
    if(n<0)
        return 0;
    if(arr[n] == k)
        return 1;
    else
        return 0;
    return linear_search(arr,n-1,k);
}

int binary_search(int arr[],int start,int end,int k)
{

```

```
if(start>end)
return 0;
int mid = (start + end )/ 2;
if(arr[mid]==k)
return 1;
else if(arr[mid]>k)
end = mid - 1;
else
start = mid + 1;
return binary_search(arr,start,end,k);
}
int main()
{
int n,i;
printf("Enter the size of array:");
scanf("%d",&n);
int choice;
int arr[n],k;
printf("\n");
printf("Enter the element to search:");
scanf("%d",&k);
while(choice!=3)
{
printf("1 - Linear search\n");
printf("2 - Binary search\n");
printf("3 - Exit\n");
printf("Enter your choice:");
scanf("%d",&choice);
if(choice==1)
```

```

{
start=clock();
for(i=0;i<80000000;i++);
int a = linear_search(arr,n-1,k);
end=clock();
if(a==1)
printf("Element found\n");
else
printf("Element not found\n");

printf("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-
start))/CLOCKS_PER_SEC));
}

else if(choice==2)
{
start=clock();
for(i=0;i<80000000;i++);
int b = binary_search(arr,0,n-1,k);
end=clock();
if(b==1)
printf("Element found\n");
else
printf("Element not found\n");

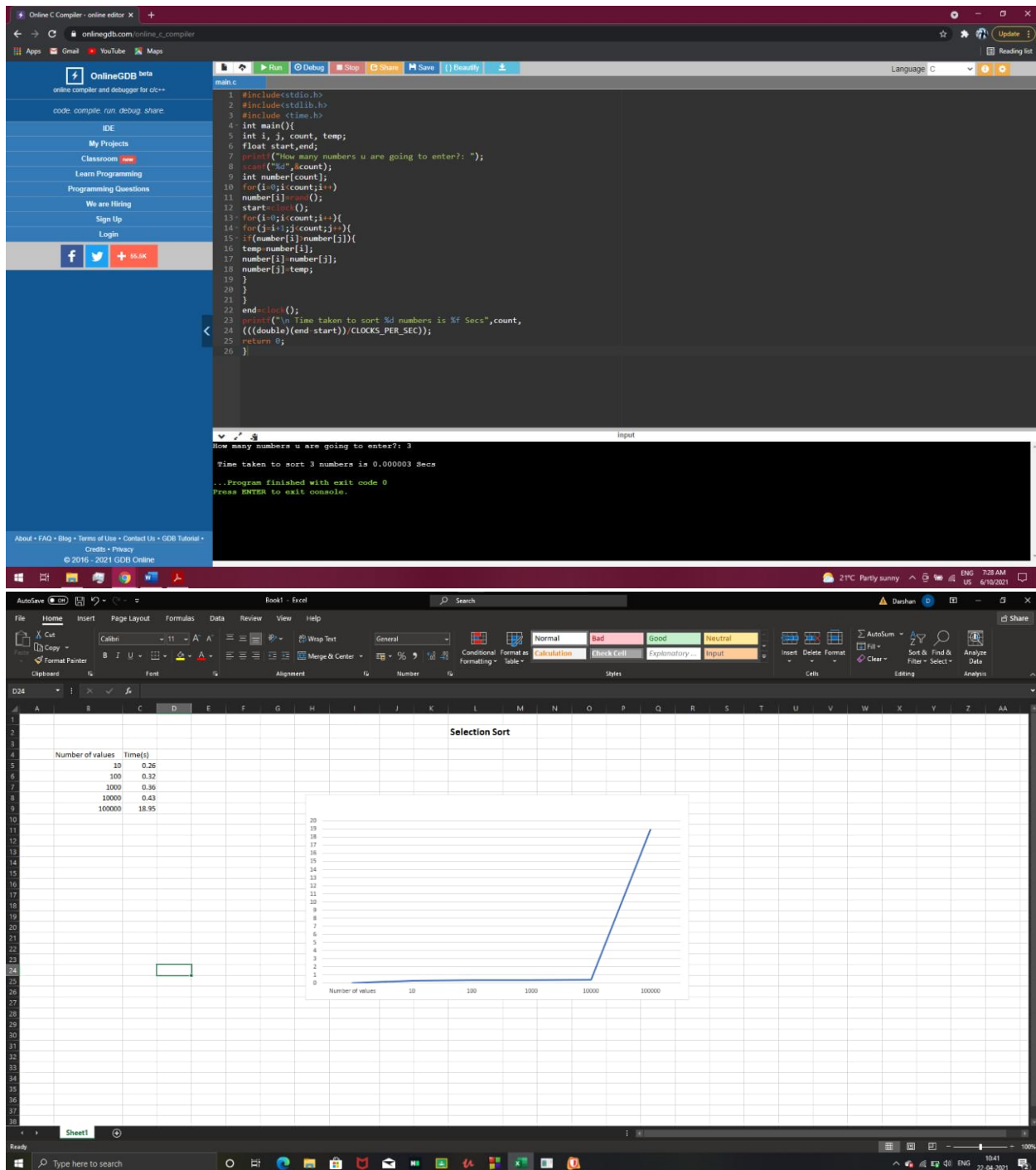
printf("\n Time taken to search value in %d numbers is %f Secs\n",n, (((double)(end-
start))/CLOCKS_PER_SEC));
}

else if(choice==3)
break;

else
printf("Invalid choice;");}}

```

## OUTPUT FOR SELECTION SORT



## CODE FOR SELECTION SORT

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include <time.h>
```



```
int main(){
int i, j, count, temp;
float start,end;
printf("How many numbers u are going to enter?: ");
scanf("%d",&count);
int number[count];
for(i=0;i<count;i++)
number[i]=rand();
start=clock();
for(i=0;i<count;i++){
for(j=i+1;j<count;j++){
if(number[i]>number[j]){
temp=number[i];
number[i]=number[j];
number[j]=temp;
}
}
}
end=clock();
printf("\n Time taken to sort %d numbers is %f Secs",count,
(((double)(end-start))/CLOCKS_PER_SEC));
return 0;
}
```