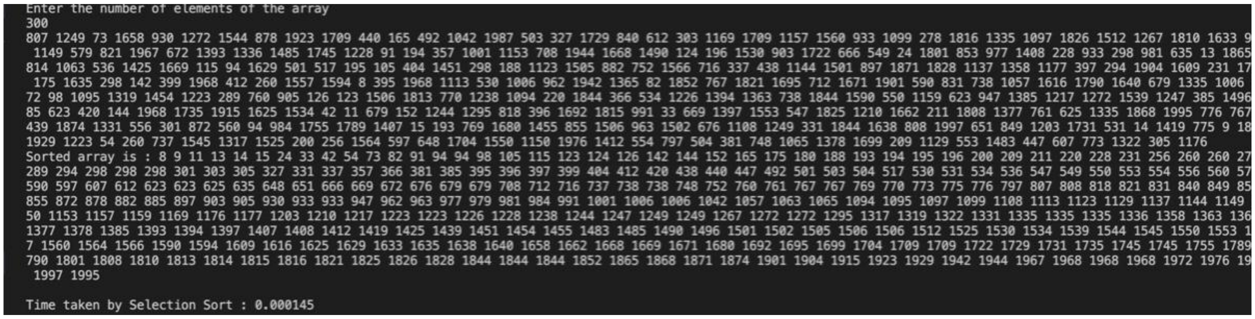
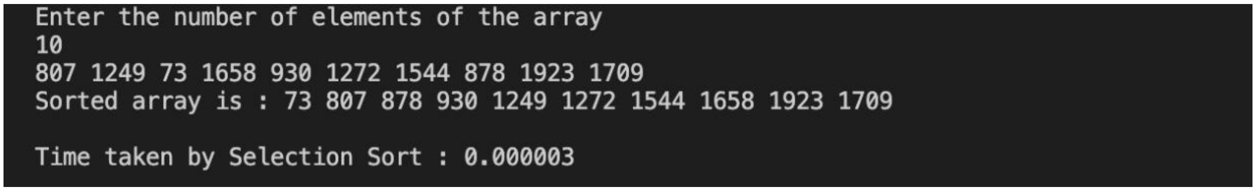
SELECTION SORT TIME

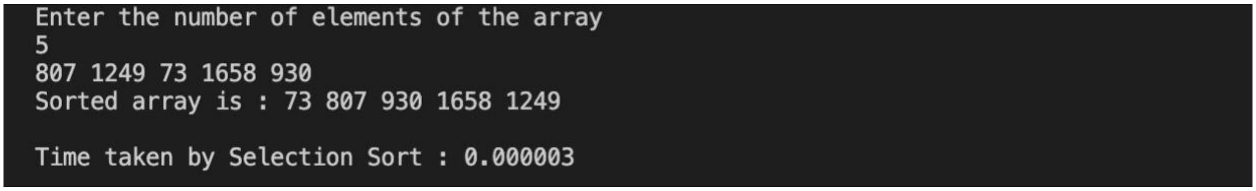
```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int n;
void selection(int a[],int n)
{


        int min,i,j,t;

        for(i=0;i<n-2;i++)
        {
                min=i;
                for(j=i+1;j<n;j++)
                {
                        if(a[j]<a[min])
                        min=j;
                }
                t=a[i];
                a[i]=a[min];
                a[min]=t;
        }
}
int main()
{
 int i;
 double st;
 clock_t start, end;
 printf("Enter the number of elements of the array\n");
 scanf("%d",&n);
 int array[n],array1[n];
 for (i = 0; i < n; i++)
 {
   array[i]= rand()%2000;
   array1[i]=array[i];
   printf("%d ", array[i]);
 }
 printf("\n");
 start = clock();
 selection(array1,n);
 end = clock();
 st = ((double) (end - start)) / CLOCKS_PER_SEC;
 printf("Sorted array is : ");
 for (i = 0; i < n; i++)
 {
   printf("%d ", array1[i]);
 }
 printf("\n");
 printf("\nTime taken by Selection Sort : %lf\n", st);
 printf("\n");
 return 0;
}
```



```
Enter the number of elements of the array
300
807 1249 73 1658 930 1272 1544 878 1923 1709 440 165 492 1042 1987 503 327 1729 840 612 303 1169 1709 1157 1560 933 1099 278 1816 1335 1097 1826 1512 1267 1810 1633 9
 1149 579 821 1967 672 1393 1336 1485 1745 1228 91 194 357 1001 1153 708 1944 1668 1490 124 196 1530 903 1722 666 549 24 1801 853 977 1408 228 933 298 981 635 13 1865
814 1063 536 1425 1669 115 94 1629 501 517 195 105 404 1451 298 188 1123 1505 882 752 1566 716 337 438 1144 1501 897 1871 1828 1137 1358 1177 397 294 1904 1609 231 17
 175 1635 298 142 399 1968 412 260 1557 1594 8 395 1968 1113 530 1006 962 1942 1365 82 1852 767 1821 1695 712 1671 1901 590 831 738 1057 1616 1790 1640 679 1335 1006
72 98 1095 1319 1454 1223 289 760 905 126 123 1506 1813 770 1238 1094 220 1844 366 534 1226 1394 1363 738 1844 1590 550 1159 623 947 1385 1217 1272 1539 1247 385 1496
85 623 420 144 1968 1735 1915 1625 1534 42 11 679 152 1244 1295 818 396 1692 1815 991 33 669 1397 1553 547 1825 1210 1662 211 1808 1377 761 625 1335 1868 1995 776 767
439 1874 1331 556 301 872 560 94 984 1755 1789 1407 15 193 769 1680 1455 855 1506 963 1502 676 1108 1249 331 1844 1638 808 1997 651 849 1203 1731 531 14 1419 775 9 18
1929 1223 54 260 737 1545 1317 1525 200 256 1564 597 648 1704 1550 1150 1976 1412 554 797 504 381 748 1065 1378 1699 209 1129 553 1483 447 607 773 1322 305 1176
Sorted array is : 8 9 11 13 14 15 24 33 42 54 73 82 91 94 94 98 105 115 123 124 126 142 144 152 165 175 180 188 193 194 195 196 200 209 211 220 228 231 256 260 260 27
289 294 298 298 298 301 303 305 327 331 337 357 366 381 385 395 396 397 399 404 412 420 438 440 447 492 501 503 504 517 530 531 534 536 547 549 550 553 554 556 560 57
590 597 607 612 623 623 625 635 648 651 666 669 672 676 679 679 708 712 716 737 738 738 748 752 760 761 767 767 769 770 773 775 776 797 807 808 818 821 831 840 849 85
855 872 878 882 885 897 903 905 930 933 933 947 962 963 977 979 981 984 991 1001 1006 1006 1042 1057 1063 1065 1094 1095 1097 1099 1108 1113 1123 1129 1137 1144 1149
50 1153 1157 1159 1169 1176 1177 1203 1210 1217 1223 1223 1226 1228 1238 1244 1247 1249 1249 1267 1272 1272 1295 1317 1319 1322 1331 1335 1335 1335 1336 1358 1363 136
1377 1378 1385 1393 1394 1397 1407 1408 1412 1419 1425 1439 1451 1454 1455 1483 1485 1490 1496 1501 1502 1505 1506 1506 1512 1525 1530 1534 1539 1544 1545 1550 1553 1
7 1560 1564 1566 1590 1594 1609 1616 1625 1629 1633 1635 1638 1640 1658 1662 1668 1669 1671 1680 1692 1695 1699 1704 1709 1709 1722 1729 1731 1735 1745 1745 1755 1789
790 1801 1808 1810 1813 1814 1815 1816 1821 1825 1826 1828 1844 1844 1844 1852 1865 1868 1871 1874 1901 1904 1915 1923 1929 1942 1944 1967 1968 1968 1968 1972 1976 19
 1997 1995

Time taken by Selection Sort : 0.000145
```



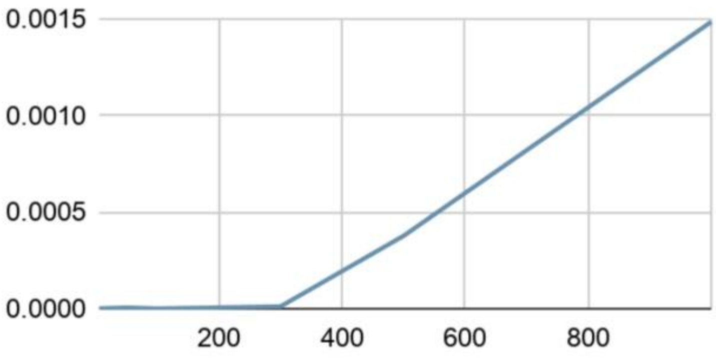
```
Enter the number of elements of the array
10
807 1249 73 1658 930 1272 1544 878 1923 1709
Sorted array is : 73 807 878 930 1249 1272 1544 1658 1923 1709

Time taken by Selection Sort : 0.000003
```



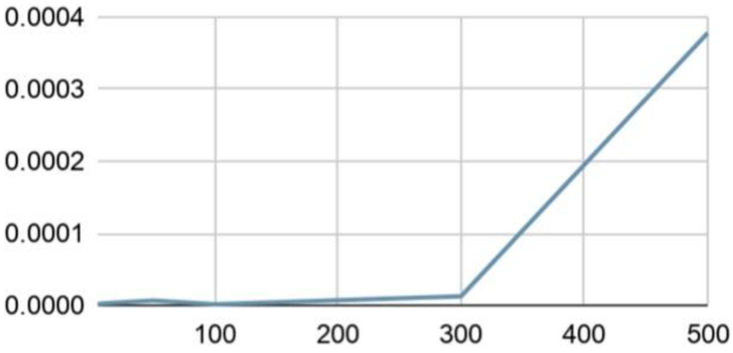
```
Enter the number of elements of the array
5
807 1249 73 1658 930
Sorted array is : 73 807 930 1658 1249

Time taken by Selection Sort : 0.000003
```

GRAPHS-

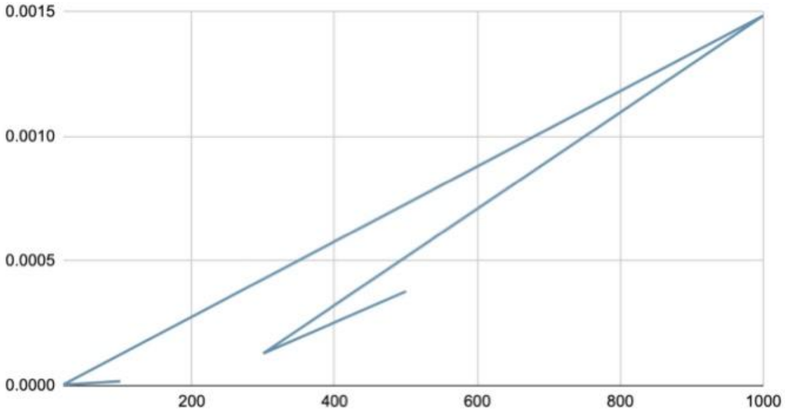
Selection sort in ascending order:



Selection sort in descending order:



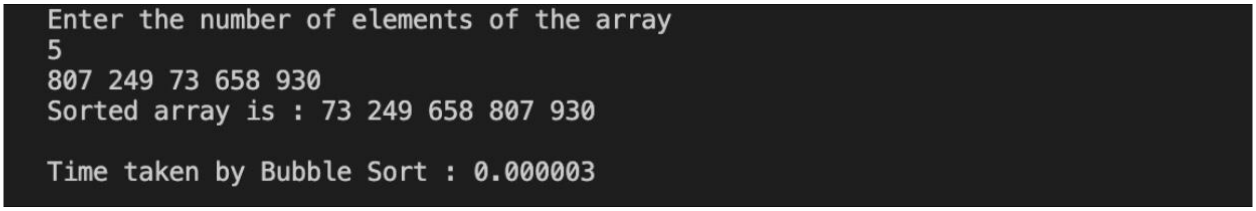
Selection sort in random order:



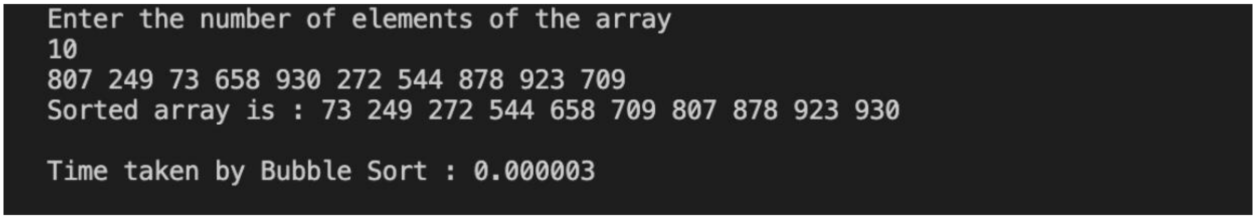
BUBBLE SORT TIME

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int n;
void swap(int *x, int *y)
{
 int temp = *x;
 *x = *y;
 *y = temp;
}
void bubbleSort(int arr[])
{
 int i, j;
 for (i = 0; i < n-1; i++)
 for (j = 0; j < n-i-1; j++)
 if (arr[j] > arr[j+1])
 swap(&arr[j], &arr[j+1]);
}
int main()
```

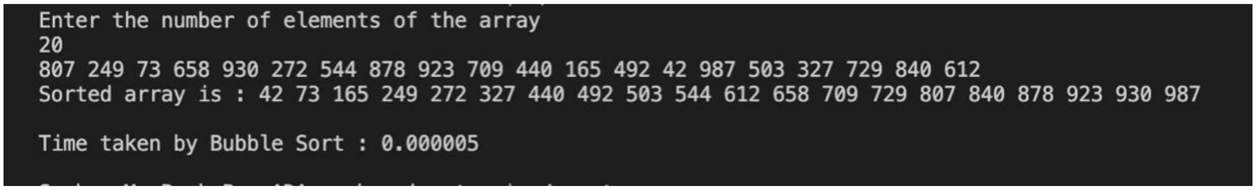
```
{
 int i;
 double bt;
 clock_t start, end;
printf("Enter the number of elements of the array\n");
 scanf("%d",&n);
 int array[n],array1[n];
 for (i = 0; i < n; i++)
 {
 array[i]= rand()%1000;
 array1[i]=array[i];
 printf("%d ", array[i]);
 }
 printf("\n");

 start = clock();
 bubbleSort(array1);
 end = clock();
 bt = ((double) (end - start)) / CLOCKS_PER_SEC;
 printf("Sorted array is : ");
 for (i = 0; i < n; i++)
 {
 printf("%d ", array1[i]);
 }
 printf("\n");
 printf("\nTime taken by Bubble Sort : %lf\n", bt);
 printf("\n");
}
```



```
Enter the number of elements of the array
5
807 249 73 658 930
Sorted array is : 73 249 658 807 930

Time taken by Bubble Sort : 0.000003
```



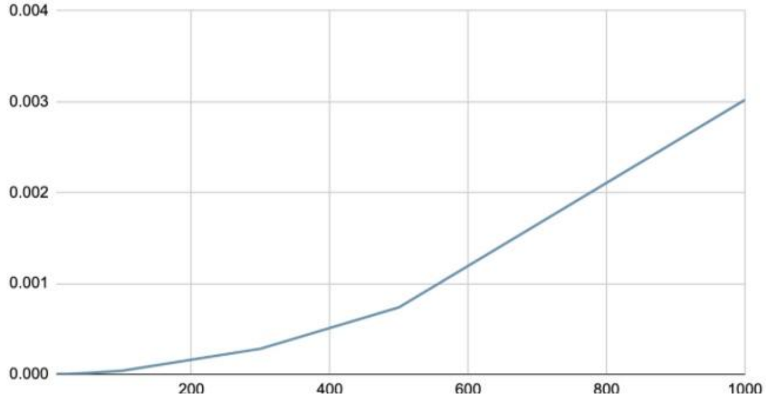
```
Enter the number of elements of the array
10
807 249 73 658 930 272 544 878 923 709
Sorted array is : 73 249 272 544 658 709 807 878 923 930

Time taken by Bubble Sort : 0.000003
```



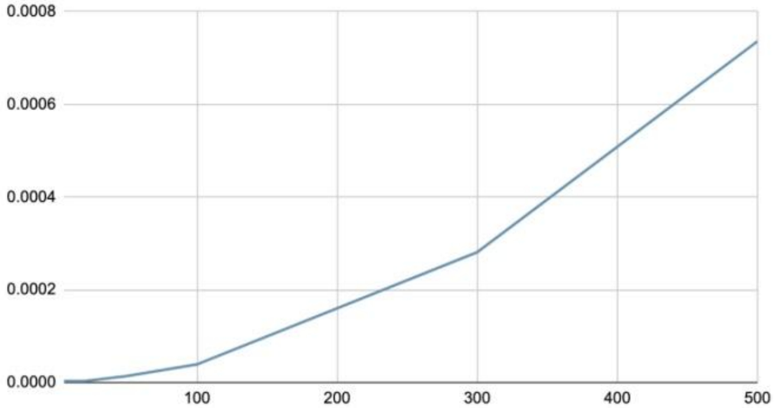
```
Enter the number of elements of the array
20
807 249 73 658 930 272 544 878 923 709 440 165 492 42 987 503 327 729 840 612
Sorted array is : 42 73 165 249 272 327 440 492 503 544 612 658 709 729 807 840 878 923 930 987

Time taken by Bubble Sort : 0.000005
```

GRAPHS-

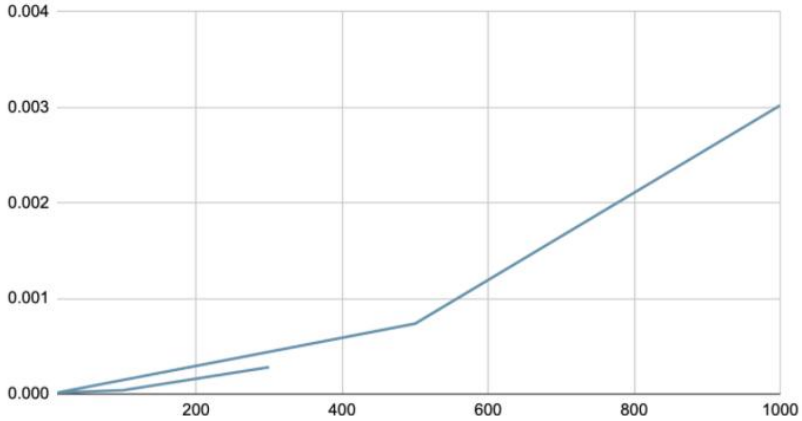
Bubble sort in ascending order:



Bubble sort in descending order:



Bubble sort in random order:


0.004
0.003
0.002
0.001
0.000
200
400
600
800
1000