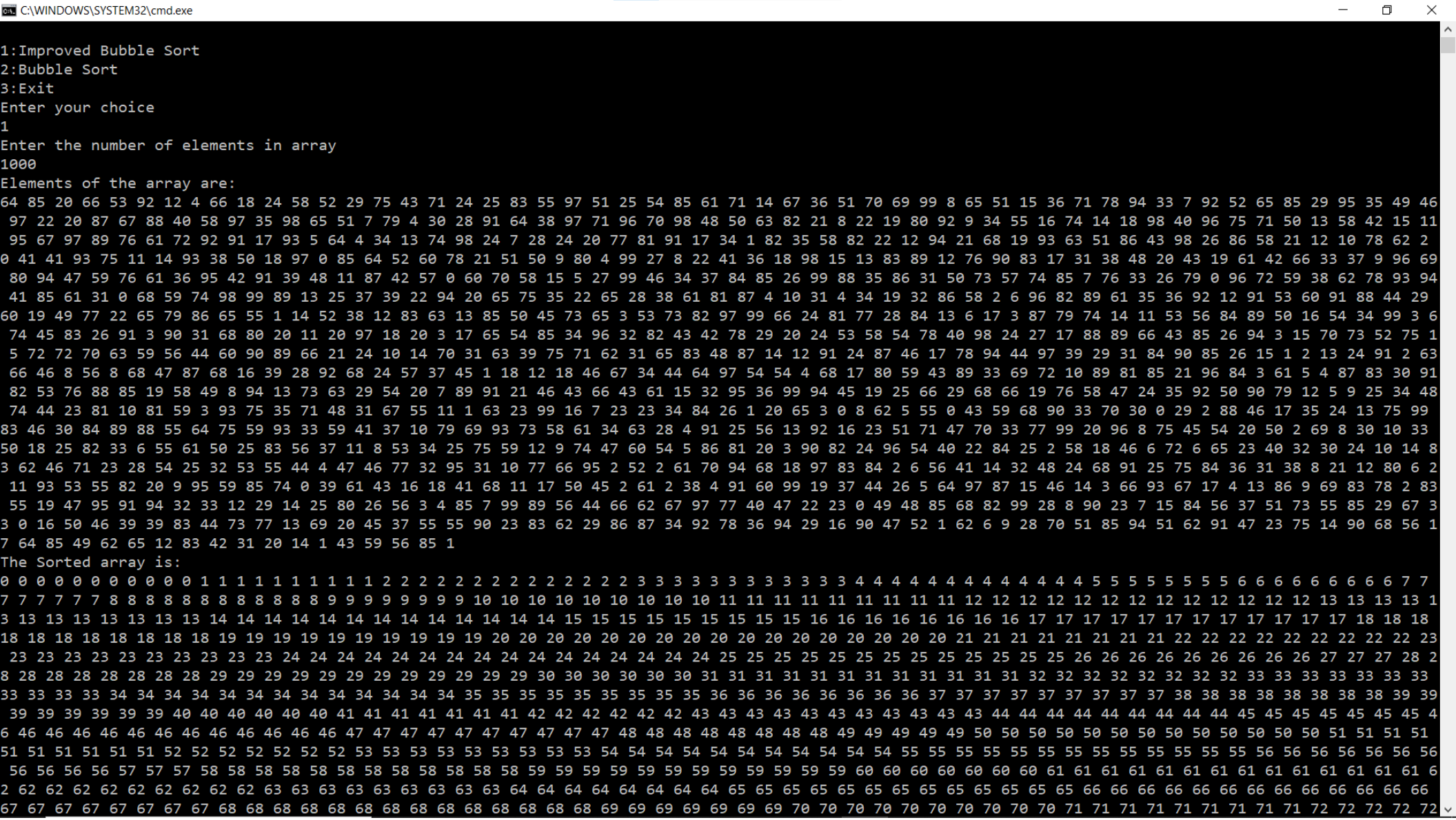
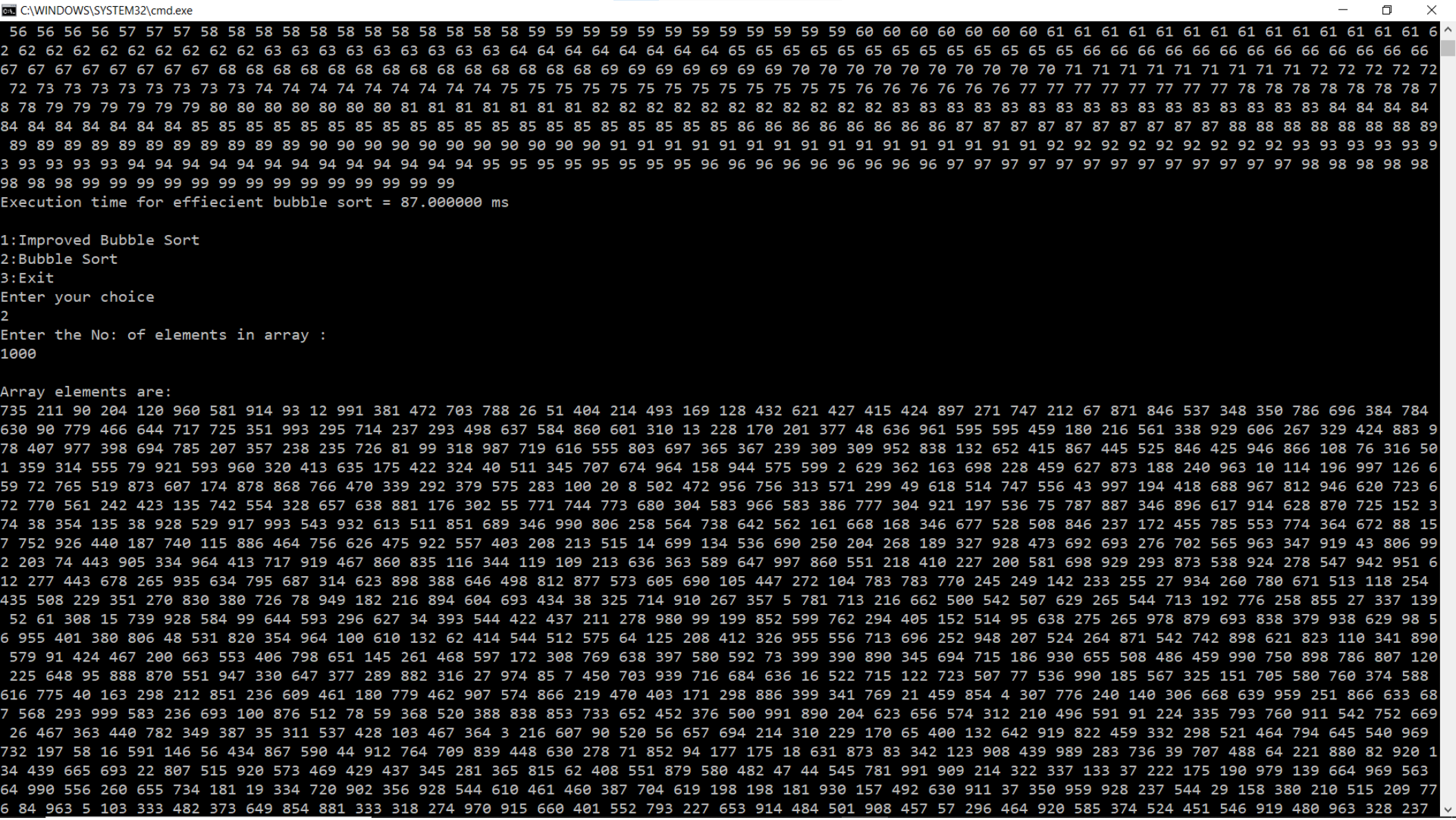
```c
#include<stdio.h>
#include <stdlib.h>
#include <time.h>
clock_t start, end;
double cpu_time;

void improved_bubble(int arr[], int n)
{
    int i,j,temp,flag;
    for(i=0;i<n-1;i++)
    {
        flag=0;
        for(j=0;j<n-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                flag=1;
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
        if(flag==0)
        {
            break;
        }
    }
    printf("\nThe Sorted array is:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
void bubble_sort(int arr[], int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
```

```
}
void bubble_sort(int arr[], int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    printf("\nThe Sorted array is:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}

int main()
{
    int i,n,c,d,k,flag=1,choice,arr[10000];
    srand(time(0));
    while(flag==1)
    {
        printf("\n1:Improved Bubble Sort\n2:Bubble Sort\n3:Exit\n");
        printf("Enter your choice\n");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1:
            printf("Enter the number of elements in array\n");
            scanf("%d", &n);
            printf("Elements of the array are:\n");
```

```
60      while(flag==1)
61      {
62          printf("\n1:Improved Bubble Sort\n2:Bubble Sort\n3:Exit\n");
63          printf("Enter your choice\n");
64          scanf("%d", &choice);
65          switch(choice)
66          {
67              case 1:
68              printf("Enter the number of elements in array\n");
69              scanf("%d", &n);
70              printf("Elements of the array are:\n");
71              for (i= 0; i<n; i++)
72              {
73                 arr[i]=rand()%100;
74                 printf("%d ",arr[i]);
75              }
76              start = clock();
77              improved_bubble(arr,n);
78              end = clock();
79              cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
80              printf("\nExecution time for the Improved bubble sort is : %f ms\n", cpu_time*1000);
81              break;
82              case 2:
83              printf("Enter the No: of elements in array :\n");
84              scanf("%d", &n);
85              printf("\nArray elements are:\n");
86              for (i= 0; i<n; i++)
87              {
88                 arr[i]=rand()%1000;
89                 printf("%d ",arr[i]);
90              }
91              start = clock();
92              bubble_sort(arr,n);
93              end = clock();
94              cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
95              printf("\nExecution time for Bubble Sort is : %f ms\n", cpu_time*1000);
96              break;
97              default:flag=0;
```

```
66          {
67              case 1:
68              printf("Enter the number of elements in array\n");
69              scanf("%d", &n);
70              printf("Elements of the array are:\n");
71              for (i= 0; i<n; i++)
72              {
73                 arr[i]=rand()%100;
74                 printf("%d ",arr[i]);
75              }
76              start = clock();
77              improved_bubble(arr,n);
78              end = clock();
79              cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
80              printf("\nExecution time for the Improved bubble sort is : %f ms\n", cpu_time*1000);
81              break;
82              case 2:
83              printf("Enter the No: of elements in array :\n");
84              scanf("%d", &n);
85              printf("\nArray elements are:\n");
86              for (i= 0; i<n; i++)
87              {
88                 arr[i]=rand()%1000;
89                 printf("%d ",arr[i]);
90              }
91              start = clock();
92              bubble_sort(arr,n);
93              end = clock();
94              cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
95              printf("\nExecution time for Bubble Sort is : %f ms\n", cpu_time*1000);
96              break;
97              default:flag=0;
98          }
99      }
100     return 0;
101 }
102
```


C:\WINDOWS\SYSTEM32\cmd.exe
1:Improved Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
1
Enter the number of elements in array
1000
Elements of the array are:
64 85 20 66 53 92 12 4 66 18 24 58 52 29 75 43 71 24 25 83 55 97 51 25 54 85 61 71 14 67 36 51 70 69 99 8 65 51 15 36 71 78 94 33 7 92 52 65 85 29 95 35 49 46
97 22 20 87 67 88 40 58 97 35 98 65 51 7 79 4 30 28 91 64 38 97 71 96 70 98 48 50 63 82 21 8 22 19 80 92 9 34 55 16 74 14 18 98 40 96 75 71 50 13 58 42 15 11
95 67 97 89 76 61 72 92 91 17 93 5 64 4 34 13 74 98 24 7 28 24 20 77 81 91 17 34 1 82 35 58 82 22 12 94 21 68 19 93 63 51 86 43 98 26 86 58 21 12 10 78 62 2
0 41 41 93 75 11 14 93 38 50 18 97 0 85 64 52 60 78 21 51 50 9 80 4 99 27 8 22 41 36 18 98 15 13 83 89 12 76 90 83 17 31 38 48 20 43 19 61 42 66 33 37 9 96 69
80 94 47 59 76 61 36 95 42 91 39 48 11 87 42 57 0 60 70 58 15 5 27 99 46 34 37 84 85 26 99 88 35 86 31 50 73 57 74 85 7 76 33 26 79 0 96 72 59 38 62 78 93 94
41 85 61 31 0 68 59 74 98 99 89 13 25 37 39 22 94 20 65 75 35 22 65 28 38 61 81 87 4 10 31 4 34 19 32 86 58 2 6 96 82 89 61 35 36 92 12 91 53 60 91 88 44 29
60 19 49 77 22 65 79 86 65 55 1 14 52 38 12 83 63 13 85 50 45 73 65 3 53 73 82 97 99 66 24 81 77 28 84 13 6 17 3 87 79 74 14 11 53 56 84 89 50 16 54 34 99 3 6
74 45 83 26 91 3 90 31 68 80 20 11 20 97 18 20 3 17 65 54 85 34 96 32 82 43 42 78 29 20 24 53 58 54 78 40 98 24 27 17 88 89 66 43 85 26 94 3 15 70 73 52 75 1
5 72 72 70 63 59 56 44 60 90 89 66 21 24 10 14 70 31 63 39 75 71 62 31 65 83 48 87 14 12 91 24 87 46 17 78 94 44 97 39 29 31 84 90 85 26 15 1 2 13 24 91 2 63
66 46 8 56 8 68 47 87 68 16 39 28 92 68 24 57 37 45 1 18 12 18 46 67 34 44 64 97 54 54 4 68 17 80 59 43 89 33 69 72 10 89 81 85 21 96 84 3 61 5 4 87 83 30 91
82 53 76 88 85 19 58 49 8 94 13 73 63 29 54 20 7 89 91 21 46 43 66 43 61 15 32 95 36 99 94 45 19 25 66 29 68 66 19 76 58 47 24 35 92 50 90 79 12 5 9 25 34 48
74 44 23 81 10 81 59 3 93 75 35 71 48 31 67 55 11 1 63 23 99 16 7 23 23 34 84 26 1 20 65 3 0 8 62 5 55 0 43 59 68 90 33 70 30 0 29 2 88 46 17 35 24 13 75 99
83 46 30 84 89 88 55 64 75 59 93 33 59 41 37 10 79 69 93 73 58 61 34 63 28 4 91 25 56 13 92 16 23 51 71 47 70 33 77 99 20 96 8 75 45 54 20 50 2 69 8 30 10 33
50 18 25 82 33 6 55 61 50 25 83 56 37 11 8 53 34 25 75 59 12 9 74 47 60 54 5 86 81 20 3 90 82 24 96 54 40 22 84 25 2 58 18 46 6 72 6 65 23 40 32 30 24 10 14 8
3 62 46 71 23 28 54 25 32 53 55 44 4 47 46 77 32 95 31 10 77 66 95 2 52 2 61 70 94 68 18 97 83 84 2 6 56 41 14 32 48 24 68 91 25 75 84 36 31 38 8 21 12 80 6 2
11 93 53 55 82 20 9 95 59 85 74 0 39 61 43 16 18 41 68 11 17 50 45 2 61 2 38 4 91 60 99 19 37 44 26 5 64 97 87 15 46 14 3 66 93 67 17 4 13 86 9 69 83 78 2 83
55 19 47 95 91 94 32 33 12 29 14 25 80 26 56 3 4 85 7 99 89 56 44 66 62 67 97 77 40 47 22 23 0 49 48 85 68 82 99 28 8 90 23 7 15 84 56 37 51 73 55 85 29 67 3
3 0 16 50 46 39 39 83 44 73 77 13 69 20 45 37 55 55 90 23 83 62 29 86 87 34 92 78 36 94 29 16 90 47 52 1 62 6 9 28 70 51 85 94 51 62 91 47 23 75 14 90 68 56 1
7 64 85 49 62 65 12 83 42 31 20 14 1 43 59 56 85 1
The Sorted array is:
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 7 7
7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 1
3 13 13 13 13 13 13 13 14 14 14 14 14 14 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17 17 17 17 17 17 18 18 18
18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22 23
23 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 25 25 25 25 25 25 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26 27 27 27 28 2
8 28 28 28 28 28 28 28 29 29 29 29 29 29 29 29 29 29 29 29 30 30 30 30 30 30 31 31 31 31 31 31 31 31 31 31 31 31 32 32 32 32 32 32 32 32 33 33 33 33 33 33 33
33 33 33 33 34 34 34 34 34 34 34 34 34 34 34 34 34 35 35 35 35 35 35 35 35 35 36 36 36 36 36 36 36 36 37 37 37 37 37 37 37 37 38 38 38 38 38 38 38 38 39 39
39 39 39 39 39 40 40 40 40 40 40 41 41 41 41 41 41 41 42 42 42 42 42 42 43 43 43 43 43 43 43 43 43 43 43 44 44 44 44 44 44 44 44 44 45 45 45 45 45 45 45 4
6 46 46 46 46 46 46 46 46 46 46 46 46 47 47 47 47 47 47 47 47 47 47 48 48 48 48 48 48 48 48 49 49 49 49 49 50 50 50 50 50 50 50 50 50 50 50 50 50 51 51 51 51
51 51 51 51 51 51 52 52 52 52 52 52 52 53 53 53 53 53 53 53 53 53 54 54 54 54 54 54 54 54 54 54 55 55 55 55 55 55 55 55 55 55 55 55 56 56 56 56 56 56 56 56
56 56 56 56 57 57 57 58 58 58 58 58 58 58 58 58 58 58 58 59 59 59 59 59 59 59 59 59 59 59 60 60 60 60 60 60 60 61 61 61 61 61 61 61 61 61 61 61 61 61 61 6
2 62 62 62 62 62 62 62 62 62 63 63 63 63 63 63 63 63 63 64 64 64 64 64 64 64 64 65 65 65 65 65 65 65 65 65 65 65 65 65 66 66 66 66 66 66 66 66 66 66 66 66
67 67 67 67 67 67 67 67 68 68 68 68 68 68 68 68 68 68 68 68 68 69 69 69 69 69 69 69 70 70 70 70 70 70 70 70 70 71 71 71 71 71 71 71 71 71 72 72 72 72 72


C:\WINDOWS\SYSTEM32\cmd.exe
56 56 56 56 57 57 57 58 58 58 58 58 58 58 58 58 58 58 58 59 59 59 59 59 59 59 59 59 59 59 59 60 60 60 60 60 60 60 61 61 61 61 61 61 61 61 61 61 61 61 62 62 62 62 62 62 62 62 62 63 63 63 63 63 63 63 63 63 64 64 64 64 64 64 64 65 65 65 65 65 65 65 65 65 65 65 65 66 66 66 66 66 66 66 66 66 66 66 66 66 67 67 67 67 67 67 67 67 68 68 68 68 68 68 68 68 68 68 68 68 68 69 69 69 69 69 69 69 70 70 70 70 70 70 70 70 70 70 71 71 71 71 71 71 71 71 71 72 72 72 72 72 72 73 73 73 73 73 73 73 73 74 74 74 74 74 74 74 74 74 75 75 75 75 75 75 75 75 75 75 75 75 75 76 76 76 76 76 76 77 77 77 77 77 77 77 77 78 78 78 78 78 78 78 78 78 79 79 79 79 79 79 80 80 80 80 80 80 80 81 81 81 81 81 81 81 82 82 82 82 82 82 82 82 82 82 82 83 83 83 83 83 83 83 83 83 83 83 83 83 83 83 84 84 84 84 84 84 84 84 84 84 84 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 86 86 86 86 86 86 86 86 87 87 87 87 87 87 87 87 87 87 88 88 88 88 88 88 88 89 89 89 89 89 89 89 89 89 89 89 90 90 90 90 90 90 90 90 90 90 90 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 92 92 92 92 92 92 92 92 92 93 93 93 93 93 93 93 93 93 94 94 94 94 94 94 94 94 94 94 94 94 94 95 95 95 95 95 95 95 95 96 96 96 96 96 96 96 96 96 97 97 97 97 97 97 97 97 97 97 97 97 98 98 98 98 98 98 98 98 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
Execution time for effiecient bubble sort = 87.000000 ms
1:Improved Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
2
Enter the No: of elements in array :
1000
Array elements are:
735 211 90 204 120 960 581 914 93 12 991 381 472 703 788 26 51 404 214 493 169 128 432 621 427 415 424 897 271 747 212 67 871 846 537 348 350 786 696 384 784 630 90 779 466 644 717 725 351 993 295 714 237 293 498 637 584 860 601 310 13 228 170 201 377 48 636 961 595 595 459 180 216 561 338 929 606 267 329 424 883 978 407 977 398 694 785 207 357 238 235 726 81 99 318 987 719 616 555 803 697 365 367 239 309 309 952 838 132 652 415 867 445 525 846 425 946 866 108 76 316 501 359 314 555 79 921 593 960 320 413 635 175 422 324 40 511 345 707 674 964 158 944 575 599 2 629 362 163 698 228 459 627 873 188 240 963 10 114 196 997 126 659 72 765 519 873 607 174 878 868 766 470 339 292 379 575 283 100 20 8 502 472 956 756 313 571 299 49 618 514 747 556 43 997 194 418 688 967 812 946 620 723 672 770 561 242 423 135 742 554 328 657 638 881 176 302 55 771 744 773 680 304 583 966 583 386 777 304 921 197 536 75 787 887 346 896 617 914 628 870 725 152 374 38 354 135 38 928 529 917 993 543 932 613 511 851 689 346 990 806 258 564 738 642 562 161 668 168 346 677 528 508 846 237 172 455 785 553 774 364 672 88 157 752 926 440 187 740 115 886 464 756 626 475 922 557 403 208 213 515 14 699 134 536 690 250 204 268 189 327 928 473 692 693 276 702 565 963 347 919 43 806 992 203 74 443 905 334 964 413 717 919 467 860 835 116 344 119 109 213 636 363 589 647 997 860 551 218 410 227 200 581 698 929 293 873 538 924 278 547 942 951 612 277 443 678 265 935 634 795 687 314 623 898 388 646 498 812 877 573 605 690 105 447 272 104 783 783 770 245 249 142 233 255 27 934 260 780 671 513 118 254 435 508 229 351 270 830 380 726 78 949 182 216 894 604 693 434 38 325 714 910 267 357 5 781 713 216 662 500 542 507 629 265 544 713 192 776 258 855 27 337 139 52 61 308 15 739 928 584 99 644 593 296 627 34 393 544 422 437 211 278 980 99 199 852 599 762 294 405 152 514 95 638 275 265 978 879 693 838 379 938 629 98 56 955 401 380 806 48 531 820 354 964 100 610 132 62 414 544 512 575 64 125 208 412 326 955 556 713 696 252 948 207 524 264 871 542 742 898 621 823 110 341 890 579 91 424 467 200 663 553 406 798 651 145 261 468 597 172 308 769 638 397 580 592 73 399 390 890 345 694 715 186 930 655 508 486 459 990 750 898 786 807 120 225 648 95 888 870 551 947 330 647 377 289 882 316 27 974 85 7 450 703 939 716 684 636 16 522 715 122 723 507 77 536 990 185 567 325 151 705 580 760 374 588 616 775 40 163 298 212 851 236 609 461 180 779 462 907 574 866 219 470 403 171 298 886 399 341 769 21 459 854 4 307 776 240 140 306 668 639 959 251 866 633 687 568 293 999 583 236 693 100 876 512 78 59 368 520 388 838 853 733 652 452 376 500 991 890 204 623 656 574 312 210 496 591 91 224 335 793 760 911 542 752 669 26 467 363 440 782 349 387 35 311 537 428 103 467 364 3 216 607 90 520 56 657 694 214 310 229 170 65 400 132 642 919 822 459 332 298 521 464 794 645 540 969 732 197 58 16 591 146 56 434 867 590 44 912 764 709 839 448 630 278 71 852 94 177 175 18 631 873 83 342 123 908 439 989 283 736 39 707 488 64 221 880 82 920 134 439 665 693 22 807 515 920 573 469 429 437 345 281 365 815 62 408 551 879 580 482 47 44 545 781 991 909 214 322 337 133 37 222 175 190 979 139 664 969 563 64 990 556 260 655 734 181 19 334 720 902 356 928 544 610 461 460 387 704 619 198 198 181 930 157 492 630 911 37 350 959 928 237 544 29 158 380 210 515 209 776 84 963 5 103 333 482 373 649 854 881 333 318 274 970 915 660 401 552 793 227 653 914 484 501 908 457 57 296 464 920 585 374 524 451 546 919 480 963 328 237

C:\WINDOWS\SYSTEM32\cmd.exe

```
34 439 665 693 22 807 515 920 573 469 429 437 345 281 365 815 62 408 551 879 580 482 47 44 545 781 991 909 214 322 337 133 37 222 175 190 979 139 664 969 563
64 990 556 260 655 734 181 19 334 720 902 356 928 544 610 461 460 387 704 619 198 198 181 930 157 492 630 911 37 350 959 928 237 544 29 158 380 210 515 209 77
6 84 963 5 103 333 482 373 649 854 881 333 318 274 970 915 660 401 552 793 227 653 914 484 501 908 457 57 296 464 920 585 374 524 451 546 919 480 963 328 237
686 282 795 16 373 265 27 59 574 434 449 271 853 187 440 350 554 384 820 539 745 220 258 912 503 401 768 48 15 614 230 94 764 616 221 766 807 21 223 898 498 6
29 944 791 734 849 235 730 842 761 87 149 430 336 36 695 71 251 849 834 939 322 752 6 623 542 676 870 391 516 161 938 855 664 957 13 47 765 933 568 948 644 43
 926 4 372 655 705 466 81 668 149 292 928 421 740 153 585 767 698 678 54 888 689 139 940 176
The Sorted array is:
2 3 4 4 5 5 6 7 8 10 12 13 13 14 15 15 16 16 16 18 19 20 21 21 22 26 26 27 27 27 27 29 34 35 36 37 37 38 38 38 39 40 40 43 43 43 44 44 47 47 48 48 48 49 51 52
 54 55 56 56 56 57 58 59 59 61 62 62 64 64 64 65 67 71 71 72 73 74 75 76 77 78 78 79 81 81 82 83 84 85 87 88 90 90 90 91 91 93 94 94 95 95 98 99 99 99 100 100
 100 103 103 104 105 108 109 110 114 115 116 118 119 120 120 122 123 125 126 128 132 132 132 133 134 134 135 135 139 139 139 140 142 145 146 149 149 151 152 1
52 153 157 157 158 158 161 161 163 163 168 169 170 170 171 172 172 174 175 175 175 176 176 177 180 180 181 181 182 185 186 187 187 188 189 190 192 194 196 197
 197 198 198 199 200 200 201 203 204 204 204 207 207 208 208 209 210 210 211 211 212 212 213 213 214 214 214 216 216 216 216 218 219 220 221 221 222 223 224 2
25 227 227 228 228 229 229 230 233 235 235 236 236 237 237 237 237 238 239 240 240 242 245 249 250 251 251 252 254 255 258 258 258 260 260 261 264 265 265 265
 265 267 267 268 270 271 271 272 274 275 276 277 278 278 278 281 282 283 283 289 292 292 293 293 293 294 295 296 296 298 298 298 299 302 304 304 306 307 308 3
08 309 309 310 310 311 312 313 314 314 316 316 318 318 320 322 322 324 325 325 326 327 328 328 329 330 332 333 333 334 334 335 336 337 337 338 339 341 341 342
 344 345 345 345 346 346 346 347 348 349 350 350 350 351 351 354 354 356 357 357 359 362 363 363 364 364 365 365 367 368 372 373 373 374 374 374 376 377 377 3
79 379 380 380 380 381 384 384 386 387 387 388 388 390 391 393 397 398 399 399 400 401 401 401 403 403 404 405 406 407 408 410 412 413 413 414 415 415 418 421
 422 422 423 424 424 424 425 427 428 429 430 432 434 434 434 435 437 437 439 439 440 440 440 443 443 445 447 448 449 450 451 452 455 457 459 459 459 459 459 4
60 461 461 462 464 464 464 466 466 467 467 467 467 468 469 470 470 472 472 473 475 480 482 482 484 486 488 492 493 496 498 498 498 500 500 501 501 502 503 507
 507 508 508 508 511 511 512 512 513 514 514 515 515 515 516 519 520 520 521 522 524 524 525 528 529 531 536 536 536 537 537 538 539 540 542 542 542 542 543 5
44 544 544 544 544 545 546 547 551 551 551 552 553 553 554 554 555 555 556 556 556 557 561 561 562 563 564 565 567 568 568 571 573 573 574 574 574 575 575 575
 579 580 580 580 581 581 583 583 583 584 584 585 585 588 589 590 591 591 592 593 593 595 595 597 599 599 601 604 605 606 607 607 609 610 610 612 613 614 616 6
16 616 617 618 619 620 621 621 623 623 623 626 627 627 628 629 629 629 629 630 630 630 631 633 634 635 636 636 636 637 638 638 638 639 642 642 644 644 644 645
 646 647 647 648 649 651 652 652 653 655 655 655 656 657 657 659 660 662 663 664 664 665 668 668 668 669 671 672 672 674 676 677 678 678 680 684 686 687 687 6
88 689 689 690 690 692 693 693 693 693 693 694 694 694 695 696 696 697 698 698 698 699 702 703 703 704 705 705 707 707 709 713 713 713 714 714 715 715 716 717
 717 719 720 723 723 725 725 726 726 730 732 733 734 734 735 736 738 739 740 740 742 742 744 745 747 747 750 752 752 752 756 756 760 760 761 762 764 764 765 7
65 766 766 767 768 769 769 770 770 771 773 774 775 776 776 776 777 779 779 780 781 781 782 783 783 784 785 785 786 786 787 788 791 793 793 794 795 795 798 803
 806 806 806 807 807 807 812 812 815 820 820 822 823 830 834 835 838 838 838 839 842 846 846 846 849 849 851 851 852 852 853 853 854 854 855 855 860 860 860 8
66 866 866 867 867 868 870 870 870 871 871 873 873 873 873 876 877 878 879 879 880 881 881 882 883 886 886 887 888 888 890 890 890 894 896 897 898 898 898 898
 902 905 907 908 908 909 910 911 911 912 912 914 914 914 915 917 919 919 919 919 920 920 920 921 921 922 924 926 926 928 928 928 928 928 928 929 929 930 930 9
32 933 934 935 938 938 939 939 940 942 944 944 946 946 947 948 948 949 951 952 955 955 956 957 959 959 960 960 961 963 963 963 963 964 964 964 966 967 969 969
 970 974 977 978 978 979 980 987 989 990 990 990 990 991 991 991 992 993 993 997 997 997 999
Execution time for Bubble Sort = 195.000000 ms

1:Improved Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
3


------------------
```