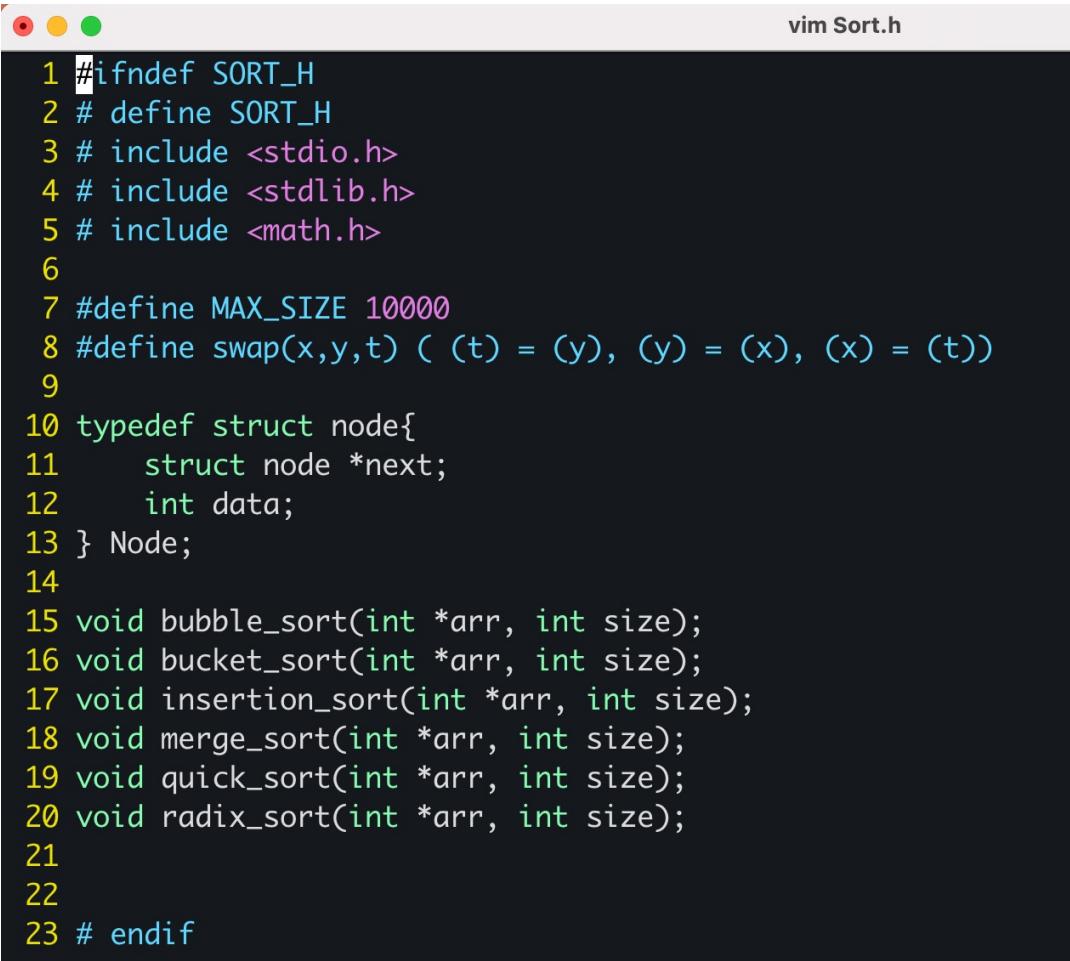


Algorithm

Midterm assignment

물리학과 20182326 이선민

Sort.h

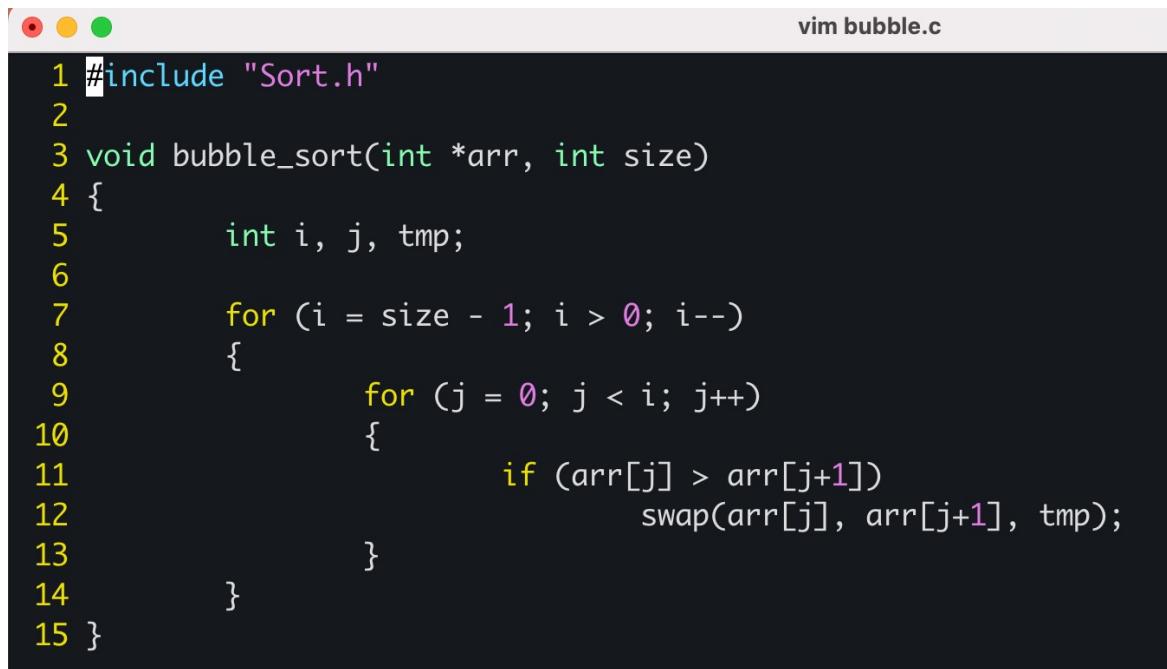


A screenshot of a terminal window showing the Vim text editor. The title bar says "vim Sort.h". The code in the editor is:

```
1 #ifndef SORT_H
2 # define SORT_H
3 # include <stdio.h>
4 # include <stdlib.h>
5 # include <math.h>
6
7 #define MAX_SIZE 10000
8 #define swap(x,y,t) ( (t) = (y), (y) = (x), (x) = (t))
9
10 typedef struct node{
11     struct node *next;
12     int data;
13 } Node;
14
15 void bubble_sort(int *arr, int size);
16 void bucket_sort(int *arr, int size);
17 void insertion_sort(int *arr, int size);
18 void merge_sort(int *arr, int size);
19 void quick_sort(int *arr, int size);
20 void radix_sort(int *arr, int size);
21
22
23 # endif
```

- The macro function used in the bubble sorting algorithm and the structure node used in the linked list implementation are separately defined in the header file here.
- The maximum size of the array is also defined here.

Bubble sort

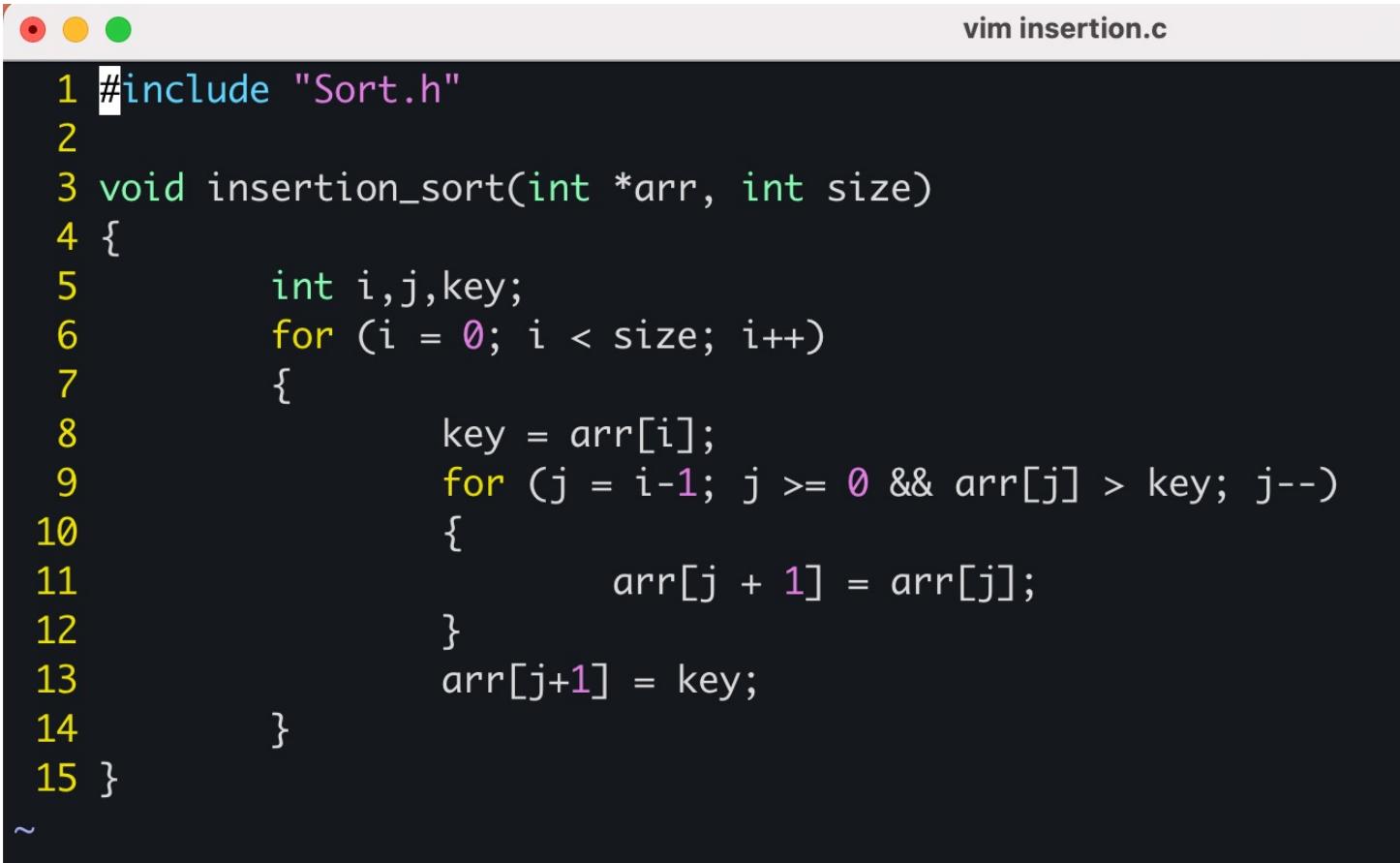


A screenshot of a terminal window showing the vim editor. The file is named 'bubble.c'. The code implements bubble sort:

```
1 #include "Sort.h"
2
3 void bubble_sort(int *arr, int size)
4 {
5     int i, j, tmp;
6
7     for (i = size - 1; i > 0; i--)
8     {
9         for (j = 0; j < i; j++)
10        {
11            if (arr[j] > arr[j+1])
12                swap(arr[j], arr[j+1], tmp);
13        }
14    }
15 }
```

- The swap function is defined in the header file.

Insertion sort



A screenshot of a terminal window showing the vim editor. The title bar says "vim insertion.c". The code in the editor is:

```
1 #include "Sort.h"
2
3 void insertion_sort(int *arr, int size)
4 {
5     int i,j,key;
6     for (i = 0; i < size; i++)
7     {
8         key = arr[i];
9         for (j = i-1; j >= 0 && arr[j] > key; j--)
10        {
11            arr[j + 1] = arr[j];
12        }
13        arr[j+1] = key;
14    }
15 }
```

The file ends with a tilde (~) at the bottom left.

Merge sort

```
1 #include "Sort.h"
2
3 void merge(int *arr, int left, int mid, int right)
4 {
5     int i = left;
6     int j = mid+1;
7     int k = left;
8     int tmp[MAX_SIZE];
9
10    while (i <= mid && j <= right)
11    {
12        if (arr[i] < arr[j])
13            tmp[k++] = arr[i++];
14        else
15            tmp[k++] = arr[j++];
16    }
17    while (i <= mid)
18        tmp[k++] = arr[i++];
19    while (j <= right)
20        tmp[k++] = arr[j++];
21    for (int a = left; a <= right; a++)
22        arr[a] = tmp[a];
23 }
24
25 void merge_recur(int *arr, int left, int right)
26 {
27     int mid = left;
28     if (left < right)
29     {
30         mid = (left+right)/2;
31         merge_recur(arr, left, mid);
32         merge_recur(arr, mid+1, right);
33     }
34     merge(arr, left, mid, right);
35 }
36
37 void merge_sort(int *arr, int size)
38 {
39     merge_recur(arr, 0, size - 1);
40 }
```

Quick sort

```
1 #include "Sort.h"
2
3 int get_pivot(int *arr, int start, int end)
4 {
5     int pivot = arr[end];
6     int tmp;
7     int i = start - 1;
8     int j = start;
9     while (j < end)
10    {
11        if (arr[j] < pivot){
12            tmp = arr[++i];
13            arr[i] = arr[j];
14            arr[j] = tmp;
15        }
16        j++;
17    }
18    tmp = arr[i+1];
19    arr[i + 1] = arr[end];
20    arr[end] = tmp;
21    return (i + 1);
22 }
23
24 void quick_recur(int *arr, int start, int end)
25 {
26     if (start >= end)
27         return ;
28     int pivot_idx;
29
30     pivot_idx = get_pivot(arr, start, end);
31     quick_recur(arr, start, pivot_idx - 1);
32     quick_recur(arr, pivot_idx + 1, end);
33 }
34
35 void quick_sort(int *arr, int size)
36 {
37     quick_recur(arr, 0, size-1);
38 }
```

Radix sort

```
1 #include "Sort.h"
2
3 void counting_sort(int *arr, int size, int *digit_arr)
4 {
5     int max = 0;
6     int i;
7     for(i = 0; i < size; i++){
8         if (arr[i] > max)
9             max = arr[i];
10    }
11    int B[MAX_SIZE];
12    int tmp[10];
13    for (i = 0; i < 10; i++)
14        tmp[i] = 0;
15    for (i = 0; i<size; i++)
16        tmp[digit_arr[i]]++;
17    for (i = 1; i<=max; i++)
18        tmp[i] = tmp[i-1] + tmp[i];
19    for (i = size - 1; i>=0; i--){
20        B[tmp[digit_arr[i]]-1] = arr[i];
21        tmp[digit_arr[i]]--;
22    }
23    for (i = 0; i<size; i++)
24        arr[i] = B[i];
25 }
26
27 int is_arr_empty(int *arr, int size)
28 {
29     for (int i = 0; i < size; i++)
30         if (arr[i] != 0)
31             return 0;
32     return 1;
33 }
34
```

```
35 void radix_sort(int *arr, int size)
36 {
37     int i;
38     int tmp[MAX_SIZE];
39     int digit_arr[MAX_SIZE];
40     int count = 0;
41     for (i = 0; i < size; i++)
42         tmp[i] = arr[i];
43     while (!is_arr_empty(tmp, size))
44     {
45         for (i = 0; i < size; i++)
46             tmp[i] = arr[i] / pow(10, count);
47         for (i = 0; i < size; i++)
48             digit_arr[i] = tmp[i] % 10;
49         count++;
50     }
51 }
52 }
```

~

Bucket sort

```
1 #include "Sort.h"
2
3 Node *Create()
4 {
5     Node *new_list = (Node *)malloc(sizeof(Node));
6     new_list->data = 0;
7     new_list->next = NULL;
8     return new_list;
9 }
10
11 void Push(Node *list, int num)
12 {
13     Node *tmp;
14     if (list->data == 0)
15     {
16         list->data = num;
17         return;
18     }
19     Node *new_node = (Node *)malloc(sizeof(Node));
20     new_node->data = num;
21     new_node->next = NULL;
22     tmp = list;
23     while (tmp->next)
24         tmp = tmp->next;
25     tmp->next = new_node;
26 }
27
28 int Pop(Node *list)
29 {
30     Node *tmp;
31     while (list->next)
32     {
33         tmp = list;
34         list = list->next;
35     }
36     tmp->next = NULL;
37     return list->data;
38 }
39
```

```
40 void sort_list(Node *list)
41 {
42     if (list->data == 0)
43         return;
44     int i,j,size,key;
45     size = 0;
46     Node *tmp = list;
47     int arr[MAX_SIZE];
48     while (tmp)
49     {
50         arr[size] = tmp->data;
51         tmp = tmp->next;
52         size++;
53     }
54
55     for (i = 0; i < size; i++)
56     {
57         key = arr[i];
58         for (j = i-1; j >= 0 && arr[j] > key; j--)
59         {
60             arr[j + 1] = arr[j];
61         }
62         arr[j+1] = key;
63     }
64     size = 0;
65     while (list)
66     {
67         list->data = arr[size++];
68         list = list->next;
69     }
70 }
```

Bucket sort

```
71
72 int list_to_arr(Node *list, int *arr, int index)
73 {
74     if (list->data == 0)
75         return index;
76     while (list)
77     {
78         arr[index++] = list->data;
79         list = list->next;
80     }
81     return index;
82 }
83
84 void free_all_list(Node *list)
85 {
86     Node *tmp;
87     while (list)
88     {
89         tmp = list;
90         list = list->next;
91         free(tmp);
92     }
93 }
94
```

```
94
95 void bucket_sort(int *arr, int size)
96 {
97     int max = arr[0];
98     int min = arr[0];
99     int index = 0;
100    int bucket_num = size / 20;
101    if (size % 20 != 0)
102        bucket_num++;
103    Node **bucket_arr = malloc(sizeof(Node *) * bucket_num);
104    if (bucket_arr == 0)
105        return;
106    int i;
107    for (i = 0; i < bucket_num; i++)
108        bucket_arr[i] = Create();
109    for (i = 0; i < size; i++)
110    {
111        if (max < arr[i])
112            max = arr[i];
113        if (min > arr[i])
114            min = arr[i];
115    }
116    int gap = (max - min + 1) / bucket_num + 1;
117    for (i = 0; i < size; i++)
118        Push(bucket_arr[arr[i] / gap], arr[i]);
119    for (i = 0; i < bucket_num; i++)
120        sort_list(bucket_arr[i]);
121    for (i = 0; i < bucket_num; i++)
122        index = list_to_arr(bucket_arr[i], arr, index);
123    for (i = 0; i < bucket_num; i++)
124        free_all_list(bucket_arr[i]);
125    free(bucket_arr);
126 }
```

main.c

```
1 #include "Sort.h"
2 #include <time.h>
3 #include <string.h>
4
5 #define ARR_SIZE1 1000
6 #define ARR_SIZE2 5000
7 #define ARR_SIZE3 10000
8
9 void init_array(int *arr, int *origin, int size)
10 {
11     int i = 0;
12     for (i = 0; i < size; i++)
13         arr[i] = origin[i];
14 }
15
16 void print_array(int *arr, int size)
17 {
18     for (int i = 0; i < size; i++)
19         printf("%d ", arr[i]);
20     printf("\n");
21 }
22
23 void random_array(int *arr, int size)
24 {
25     int i,j;
26     srand(time(NULL));
27     for (i = 0; i < size; i++)
28     {
29         arr[i] = rand()% size + 1;
30         for (j = 0; j < i; j++)
31             if (arr[i] == arr[j])
32                 i--;
33     }
34 }
35
36 void reverse_array(int *arr, int size)
37 {
38     int i;
39     for (i = 0; i < size; i++)
40         arr[i] = size - i;
41 }
42
43
44 double check_time(int *arr, int *origin, int size, void (*fp)(int *, int))
45 {
46     clock_t start, end;
47     start = clock();
48     fp(arr, size);
49     end = clock();
50     double runtime = (double)(end - start);
51     //if (size == 10000)
52     //    print_array(arr, size);
53     printf("%d size array execution time : %.0lf ms \n", size, runtime);
54     init_array(arr, origin, size);
55     return runtime;
56 }
57
58 int number_of_digits(int n)
59 {
60     if(n == 0){
61         return 0;
62     }
63     while(n != 0)
64     {
65         return 1 + number_of_digits(n/10);
66     }
67     return 0;
68 }
69
```

Main.c

```
70 void print_result(double bubble, double bucket, double insertion, double merge, double quick, double radix)
71 {
72     printf(" %.0lf ms", bubble);
73     for (int i = 0; i < (strlen("bubble sort") - 2 - number_of_digits((int)bubble)); i++)
74         printf(" ");
75     printf("|\n");
76     printf(" %.0lf ms", bucket);
77     for (int i = 0; i < (strlen("bucket sort") - 2 - number_of_digits((int)bucket)); i++)
78         printf(" ");
79     printf("|\n");
80     printf(" %.0lf ms", insertion);
81     for (int i = 0; i < (strlen("insertion sort") - 2 - number_of_digits((int)insertion)); i++)
82         printf(" ");
83     printf("|\n");
84     printf(" %.0lf ms", merge);
85     for (int i = 0; i < (strlen("merge sort") - 2 - number_of_digits((int)merge)); i++)
86         printf(" ");
87     printf("|\n");
88     printf(" %.0lf ms", quick);
89     for (int i = 0; i < (strlen("quick sort") - 2 - number_of_digits((int)quick)); i++)
90         printf(" ");
91     printf("|\n");
92     printf(" %.0lf ms", radix);
93     for (int i = 0; i < (strlen("radix sort") - 2 - number_of_digits((int)radix)); i++)
94         printf(" ");
95     printf("|\n");
96 }
```

Main.c

```
98 int main(int argc, char **argv)
99 {
100     int arr1[ARR_SIZE1];
101     int arr2[ARR_SIZE2];
102     int arr3[ARR_SIZE3];
103     int origin1[ARR_SIZE1];
104     int origin2[ARR_SIZE2];
105     int origin3[ARR_SIZE3];
106
107
108     if (argc != 1 && strcmp(argv[1], "random") == 0)
109     {
110         printf("random array\n");
111         random_array(arr1, ARR_SIZE1);
112         random_array(arr2, ARR_SIZE2);
113         random_array(arr3, ARR_SIZE3);
114     }
115     else
116     {
117         printf("reverse array\n");
118         reverse_array(arr1, ARR_SIZE1);
119         reverse_array(arr2, ARR_SIZE2);
120         reverse_array(arr3, ARR_SIZE3);
121     }
122     init_array(origin1, arr1, ARR_SIZE1);
123     init_array(origin2, arr2, ARR_SIZE2);
124     init_array(origin3, arr3, ARR_SIZE3);
125
126
127     //printf("original arr\n");
128     //print_array(arr1, ARR_SIZE1);
129     //printf("\n arr2\n");
130     //print_array(arr2, ARR_SIZE2);
131     //printf("\n arr3\n");
132     //print_array(arr3, ARR_SIZE3);
133
134     puts(" \n * bubble sort \n ");
135     double bubble1 = check_time(arr1, origin1, ARR_SIZE1, bubble_sort);
136     double bubble2 = check_time(arr2, origin2, ARR_SIZE2, bubble_sort);
137     double bubble3 = check_time(arr3, origin3, ARR_SIZE3, bubble_sort);
138
139     puts(" \n * bucket sort \n ");
140     double bucket1 = check_time(arr1, origin1, ARR_SIZE1, bucket_sort);
141     double bucket2 = check_time(arr2, origin2, ARR_SIZE2, bucket_sort);
142     double bucket3 = check_time(arr3, origin3, ARR_SIZE3, bucket_sort);
143
144     puts(" \n * insertion sort \n ");
145     double insertion1 = check_time(arr1, origin1, ARR_SIZE1, insertion_sort);
146     double insertion2 = check_time(arr2, origin2, ARR_SIZE2, insertion_sort);
147     double insertion3 = check_time(arr3, origin3, ARR_SIZE3, insertion_sort);
148
149
150     puts(" \n * merge sort \n ");
151     double merge1 = check_time(arr1, origin1, ARR_SIZE1, merge_sort);
152     double merge2 = check_time(arr2, origin2, ARR_SIZE2, merge_sort);
153     double merge3 = check_time(arr3, origin3, ARR_SIZE3, merge_sort);
154
155     puts(" \n * quick sort \n ");
156     double quick1 = check_time(arr1, origin1, ARR_SIZE1, quick_sort);
157     double quick2 = check_time(arr2, origin2, ARR_SIZE2, quick_sort);
158     double quick3 = check_time(arr3, origin3, ARR_SIZE3, quick_sort);
159
160     puts(" \n * radix sort \n ");
161     double radix1 = check_time(arr1, origin1, ARR_SIZE1, radix_sort);
162     double radix2 = check_time(arr2, origin2, ARR_SIZE2, radix_sort);
163     double radix3 = check_time(arr3, origin3, ARR_SIZE3, radix_sort);
164
165
```

Main.c

```
165
166     char str[10] = "reverse";
167     if (argc > 1 && !strcmp(argv[1], "random"))
168         strcpy(str, "random");
169     printf("\nResult Table (when %s array)\n", str);
170     printf("-----+-----+-----+-----+-----+-----+\n");
171     printf(" | bubble sort | bucket sort | insertion sort | merge sort | quick sort | radix sort |\n");
172     printf("-----+-----+-----+-----+-----+-----+\n");
173     printf(" |1000 |");
174     print_result(bubble1, bucket1, insertion1, merge1, quick1, radix1);
175     printf("-----+-----+-----+-----+-----+-----+\n");
176     printf(" |5000 |");
177     print_result(bubble2, bucket2, insertion2, merge2, quick2, radix2);
178     printf("-----+-----+-----+-----+-----+-----+\n");
179     printf(" |10000|");
180     print_result(bubble3, bucket3, insertion3, merge3, quick3, radix3);
181     printf("-----+-----+-----+-----+-----+-----+\n");
182 }
183
```

Input data – 1000 size random array

```
[~/2022_1/Algorithm/midterm]$ ./program random
random array
original arr
* [master]
271 584 928 172 550 551 564 871 972 715 368 430 392 924 290 445 80 440 257 805 508 936 435 730 224 110 558 64 768 425 363 880 948 840 671 630 939 595 910 734 227 904 528 565 567 722 982
429 273 673 480 921 500 787 946 44 324 552 460 888 115 966 166 519 377 850 878 135 708 728 360 785 803 869 855 553 664 441 301 844 744 717 59 266 923 239 75 420 746 684 452 995 304 178
82 706 11 79 694 669 99 279 815 179 255 283 22 83 862 510 520 413 485 129 222 40 889 620 957 867 772 578 237 33 874 776 962 474 245 423 616 947 327 579 217 294 444 858 650 852 216 753
27 6 779 766 541 847 185 131 73 412 568 81 903 235 687 5 752 72 732 504 780 119 792 20 925 173 761 280 663 114 268 984 66 155 467 104 108 42 35 589 57 397 999 15 309 449 879 843 287 679
899 118 814 985 594 493 788 641 160 817 799 628 902 323 63 585 387 883 307 935 264 18 774 231 259 602 164 469 161 922 727 421 330 560 634 499 69 263 472 950 200 50 934 136 311 534 546
680 142 351 800 463 848 824 667 229 906 764 384 329 582 603 777 56 278 306 873 828 543 635 659 125 4 545 951 389 919 911 554 653 599 175 111 775 396 194 672 109 632 542 868 809 94 577 5
4 437 514 395 610 117 241 758 225 198 900 496 47 731 230 481 955 841 149 367 859 993 214 735 51 208 920 305 295 601 369 297 729 188 492 284 242 102 905 252 765 631 89 163 971 544 240 52
3 388 196 45 655 832 842 587 315 381 970 58 709 34 228 991 372 341 502 253 171 887 700 570 849 916 181 702 13 498 300 137 968 983 938 364 3 676 400 825 386 379 795 829 853 835 123 705 9
98 556 84 352 340 31 433 942 592 639 580 783 293 106 189 417 37 614 244 581 645 678 933 793 583 206 716 159 637 856 697 791 141 201 414 234 529 914 398 205 743 277 931 767 701 457 285 8
31 70 737 576 223 547 180 703 629 436 912 428 375 182 674 836 626 167 361 427 760 538 723 617 85 784 559 690 789 790 177 711 978 693 834 269 940 226 692 443 328 657 468 314 71 68 370 50
1 331 975 24 376 932 8 325 929 100 810 132 539 575 140 839 450 289 320 704 969 490 813 191 113 276 419 415 41 373 344 531 604 447 548 740 833 1000 649 134 105 486 385 846 322 286 961 10
212 675 434 908 130 960 967 357 195 907 332 854 794 526 150 819 699 393 497 965 199 30 872 683 806 820 974 593 219 174 532 830 623 356 590 941 202 782 246 439 26 53 96 574 282 573 877
651 265 625 321 661 521 267 336 7 707 218 757 146 990 62 647 698 691 272 197 588 827 812 870 741 882 654 98 291 895 16 154 442 507 770 913 636 411 38 446 55 190 535 462 358 17 607 488 4
65 773 424 394 917 165 333 652 203 274 464 608 299 138 143 326 688 755 666 943 980 121 353 495 964 762 107 930 638 484 9 977 14 308 949 826 210 409 851 953 738 613 597 525 769 97 811 89
7 470 418 21 863 875 383 807 685 213 816 339 612 713 644 646 739 95 561 382 478 362 748 976 39 122 221 77 515 186 335 605 438 127 261 23 426 482 533 410 778 837 956 349 176 28 822 406 6
19 797 204 994 298 390 43 720 719 456 317 120 527 342 656 250 238 563 751 668 658 756 302 714 169 416 466 600 334 168 32 513 318 954 46 643 350 91 451 337 992 606 927 156 281 338 404 88
1 483 36 721 689 890 343 896 864 660 944 540 901 996 712 133 88 598 65 29 49 642 162 503 459 407 615 988 249 979 516 247 153 989 491 865 845 759 139 665 802 823 157 87 471 952 90 987 74
2 260 126 749 566 236 622 192 893 648 537 536 804 207 453 454 170 726 2 681 796 158 725 861 724 745 886 52 296 401 145 572 945 220 786 319 860 511 403 211 19 366 60 524 25 151 144 808 5
91 918 348 892 374 586 254 838 937 781 549 391 61 232 571 270 347 116 884 12 310 101 1 876 557 345 973 530 494 476 473 997 371 346 633 517 487 505 76 736 771 399 359 898 209 184 798 355
262 233 124 747 640 74 431 354 518 754 448 695 408 627 986 183 621 365 187 112 857 885 750 963 103 522 461 801 147 624 402 78 479 682 258 562 292 380 128 313 458 275 821 477 312 152 86
92 763 256 475 569 710 959 193 405 316 93 718 48 909 148 512 677 958 251 455 215 891 555 915 611 422 696 618 378 432 243 506 981 670 489 609 866 662 926 288 894 818 67 509 686 248 733
303 596
```

Output data - 1000 size random array

```
* bubble sort

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9
99 1000

1000 size array execution time : 1126 ms
```

Output data - 1000 size random array

```
* bucket sort

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9
99 1000

1000 size array execution time : 92 ms
```

Output data - 1000 size random array

```
* insertion sort

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9
99 1000

1000 size array execution time : 409 ms
```

Output data - 1000 size random array

```
* merge sort

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9
99 1000

1000 size array execution time : 90 ms
```

Output data - 1000 size random array

```
* quick sort

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9
99 1000

1000 size array execution time : 57 ms
```

Output data - 1000 size random array

* radix sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582  
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6  
29 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 67  
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721  
722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767  
768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 8  
14 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 86  
0 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906  
907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952  
953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 9  
99 1000
```

1000 size array execution time : 94 ms

Input data - 5000 size random array

```
[~/2022_1/Algorithm/midterm]$ ./program random
random array
original arr
2262 2972 4988 1681 362 623 2108 3129 3668 519 3629 2973 2166 471 152 3500 4616 2829 4718 3763 1489 1889 758 1652 706 2049 3936 2933 4565 4007 975 4641 1906 4835 2804 704 1229 4340 1299
94 636 3875 4123 3507 1625 3813 2058 3927 3371 3486 915 2454 3191 3545 2423 674 3553 4711 1712 4087 4638 3297 170 2246 2417 81 1272 372 921 2730 2502 1609 1233 1546 2399 687 1 2601 296
4461 1468 2345 1366 1376 113 3565 1491 4338 3917 224 266 4373 3989 1187 2268 1786 92 4239 3523 1587 3725 3966 4862 752 4983 3941 1359 4946 3642 338 4459 179 1577 1880 2463 684 3548 229
0 2331 3680 4560 459 4709 2205 3337 2862 947 2581 4481 4457 114 4398 943 4657 3289 4605 1605 1377 750 518 2116 492 4536 1262 36 4024 4971 1898 2944 747 3185 4438 444 2270 1418 3535 2899
80 1709 2476 210 4569 3903 1846 3298 1814 4728 4640 2670 1640 4045 2580 1685 759 3009 1511 1917 1472 3336 4823 1261 3538 4508 1406 1136 1799 4247 566 1995 199 368 3744 3331 2277 4860 3
690 3816 4 1309 3182 1654 2338 3826 1430 1355 2729 1020 3748 4230 4784 1943 912 365 1001 1318 140 64 1734 1237 2589 1310 3617 4514 1170 1033 4304 1970 924 3772 2440 245 3611 2046 839 31
30 1953 1414 1967 4951 289 4252 3041 1760 4917 3273 3746 21 2713 460 669 2639 1323 699 890 3882 1911 550 4114 4216 2732 3959 1466 2346 1988 2675 1196 3417 3722 2789 1797 2718 4521 1599
2959 2655 703 3055 1890 4821 1939 710 1416 2845 547 4716 2653 4977 2620 1266 2807 1121 2971 2335 348 3626 3113 3982 3527 2062 486 4858 1126 1449 3740 3646 2528 1228 137 4085 1153 4086 7
44 2233 1688 2478 702 2765 1398 1182 4610 2751 496 3908 4774 2638 2284 3295 4306 4507 2311 1142 222 2693 525 726 2144 4208 680 917 3528 3561 4228 3658 1340 1302 3096 4528 2427 260 1666
2056 461 2970 2342 2001 4162 2243 3526 4630 339 3525 286 3551 4693 148 2207 1481 3531 2563 3271 4650 1089 1068 1759 3223 4701 3186 4485 241 4041 1458 2510 4612 3881 1129 4305 1385 976 4
189 2771 4235 1439 2858 1674 1804 3841 1454 608 2868 3802 1147 633 4009 2459 3594 1815 3798 3109 4330 3384 167 994 2377 2863 2833 2849 2161 2994 454 4517 4289 3831 1086 1791 2864 714 34
61 214 2315 3357 3277 160 402 2484 3421 4034 2890 2654 1550 842 3618 3918 1018 2640 1533 422 2742 708 415 380 977 2764 4337 4941 2544 489 3914 27 4953 4154 798 3704 2095 1742 3485 3844
1405 3832 3487 3803 2778 2759 4152 3514 2293 1828 4190 2623 4885 4378 869 989 1819 2048 1955 4861 971 1689 2908 2993 3509 3448 4422 959 3260 2588 2855 1649 3577 2679 717 4928 3342 3237
2073 3368 2162 3369 4295 2700 3800 4210 3299 4105 559 1079 4139 2461 2809 3977 1235 336 1677 709 4551 1443 530 2683 1526 3162 3892 3169 2203 4794 1069 3395 1527 4070 1293 2336 3245 139
3343 697 3928 3520 3179 3698 1523 4072 4059 4362 4273 3097 275 3451 4704 2210 1171 1283 1098 2195 728 3834 1842 4806 598 637 4000 2358 1333 2665 3683 3427 2107 4822 2219 2416 4186 4286
1184 361 3301 395 332 329 2450 3779 1002 2185 3660 2495 16 4606 2451 3439 1096 1222 3136 3092 761 3924 3759 3750 4696 2050 2334 1693 1107 3776 4648 4664 1886 330 2818 2547 4965 4959 448
3 2022 4620 1369 1617 4757 3241 141 3930 3761 1463 1971 345 1764 2572 1499 1870 1687 4607 4960 1451 323 4601 1419 3516 1065 2157 4944 3878 4159 4420 743 1227 1201 4170 3422 3454 3146 42
90 2610 2937 2539 1806 1884 2938 3416 1720 4344 4264 965 3453 1177 3669 1330 3350 843 2574 2075 2131 3745 1012 575 473 490 2869 3922 2838 3689 905 2741 3476 4849 432 3479 4681 4042 3580
3084 83 1276 987 3893 407 1138 2231 4386 4166 3807 2637 1021 2480 1365 1517 3838 3089 946 1991 1164 815 466 1165 3635 35 2659 793 908 4715 396 541 3100 4064 1429 2695 1964 755 1158 667
318 4873 4039 810 3717 2721 4090 712 2615 3317 4903 2199 3585 4538 4115 40 172 828 1745 1144 4546 4981 2380 3064 628 900 1111 1963 145 4857 1249 1074 2499 1264 3639 765 4456 2609 2028
3006 1717 2834 601 2836 846 499 2374 1610 2817 4744 3218 3043 2748 3253 4945 3493 2664 178 3244 4952 1700 2255 2985 1257 2805 204 3257 3194 3228 129 3441 3656 1897 1156 816 1477 4224 27
94 4870 4057 3628 1597 3979 3897 25 3863 1178 3344 4901 935 787 771 4934 39 4428 3691 721 4054 4296 1772 817 2846 2881 3522 4129 4184 966 1558 2988 591 182 3742 2264 929 95 774 4150 135
8 4237 4531 4683 3709 3984 2477 3991 2928 1103 4578 3 3370 3792 3188 4561 498 1368 1581 605 1382 3042 1348 1277 2569 4777 3868 1514 4698 1210 2558 480 533 1322 2384 3663 1411 3413 3771
772 2398 2773 4661 1633 1485 1538 1354 3377 3076 168 1753 2920 2545 1803 4982 11 2716 1081 1563 1128 3940 4943 2230 1852 1320 1327 1231 3038 4484 1743 2397 1621 622 270 1924 1559 4293 3
322 741 672 4726 1294 4487 1529 910 4118 1978 893 5 1367 399 4689 3425 4151 642 3388 2488 2954 2288 3392 2719 28 4329 2534 2177 3751 2160 1719 3886 3632 1935 4396 4440 2493 884 1904 715
3842 381 2613 4767 1102 2887 313 443 2198 2294 4372 691 45 822 4369 1049 2241 3144 3641 3608 1879 3760 1087 4268 2467 4573 4804 3700 1660 2509 1192 2285 2749 2955 3631 2582 2978 1194 4
488 100 3123 3944 3657 725 2156 3650 1792 4022 1400 852 3840 4660 4745 481 2527 4394 2976 3814 2591 3304 2746 4710 4997 4906 3518 156 315 3786 1977 2354 2274 259 4608 4914 3235 254 1048
2273 4847 3684 2004 4926 2703 555 2059 1541 1056 568 1402 2209 523 4623 1810 887 2265 540 4958 1058 603 805 2009 4237 2625 2626 2232 2627 246 4454 2148 104 4122 2006 4
```

Output data - 5000 size random array

```
* bubble sort
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444  
445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582  
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629  
630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675  
5 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721  
  
88 4589 4590 4591 4592 4593 4594 4595 4596 4597 4598 4599 4600 4601 4602 4603 4604 4605 4606 4607 4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618 4619 4620 4621 4622 4623 4624 4625  
25 4626 4627 4628 4629 4630 4631 4632 4633 4634 4635 4636 4637 4638 4639 4640 4641 4642 4643 4644 4645 4646 4647 4648 4649 4650 4651 4652 4653 4654 4655 4656 4657 4658 4659 4660 4661 4662  
4663 4664 4665 4666 4667 4668 4669 4670 4671 4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682 4683 4684 4685 4686 4687 4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698 4699  
99 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735 4736  
4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 4773  
73 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810  
10 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 4847  
4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 4884  
84 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 4921  
4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958  
4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995  
95 4996 4997 4998 4999 5000  
5000 size array execution time : 32356 ms
```

Output data - 5000 size random array

* bucket sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259  
260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305  
306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444  
445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582  
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629  
630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675  
676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721  
  
14 4515 4516 4517 4518 4519 4520 4521 4522 4523 4524 4525 4526 4527 4528 4529 4530 4531 4532 4533 4534 4535 4536 4537 4538 4539 4540 4541 4542 4543 4544 4545 4546 4547 4548 4549 4550 4551  
4552 4553 4554 4555 4556 4557 4558 4559 4560 4561 4562 4563 4564 4565 4566 4567 4568 4569 4570 4571 4572 4573 4574 4575 4576 4577 4578 4579 4580 4581 4582 4583 4584 4585 4586 4587 4588  
4589 4590 4591 4592 4593 4594 4595 4596 4597 4598 4599 4600 4601 4602 4603 4604 4605 4606 4607 4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618 4619 4620 4621 4622 4623 4624 4625  
4626 4627 4628 4629 4630 4631 4632 4633 4634 4635 4636 4637 4638 4639 4640 4641 4642 4643 4644 4645 4646 4647 4648 4649 4650 4651 4652 4653 4654 4655 4656 4657 4658 4659 4660 4661 4662  
4663 4664 4665 4666 4667 4668 4669 4670 4671 4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682 4683 4684 4685 4686 4687 4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698 4699  
4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735 4736  
4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 4773  
4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810  
4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 4847  
4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 4884  
4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 4921  
4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958  
4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995  
4996 4997 4998 4999 5000  
5000 size array execution time : 380 ms
```

Output data - 5000 size random array

```
* insertion sort
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
```

```
62 4663 4664 4665 4666 4667 4668 4669 4670 4671 4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682 4683 4684 4685 4686 4687 4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698 4699 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735 4736 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 4773 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 4847 4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 4884 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 4921 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958 4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995 4996 4997 4998 4999 5000
```

5000 size array execution time : 9727 ms

Output data - 5000 size random array

* merge sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
  
36 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 47  
73 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 48  
10 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 48  
47 4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 48  
84 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 49  
21 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 49  
58 4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 49  
95 4996 4997 4998 4999 5000  
5000 size array execution time : 509 ms
```

Output data - 5000 size random array

* quick sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
99 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735 47  
36 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 47  
73 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 48  
10 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 48  
47 4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 48  
84 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 49  
21 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 49  
58 4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 49  
95 4996 4997 4998 4999 5000  
5000 size array execution time : 327 ms
```

Output data – 5000 size random array

* radix sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
99 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735 47  
36 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 47  
73 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 48  
10 4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 48  
47 4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882 4883 48  
84 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 49  
21 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 49  
58 4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 49  
95 4996 4997 4998 4999 5000  
5000 size array execution time : 416 ms
```

Input data - 10000 size random array

```
[~/2022_1/Algorithm/midterm]$ ./program random
random array
original arr
4333 9394 3717 9016 1623 5616 2425 254 4678 7719 5147 8953 4618 9146 766 771 6814 1390 3083 9591 5742 3161 6468 2384 1904 404 6347 5786 7354 3419 9820 5097 2092 7171 8540 8442 5075 3850
3318 3704 8142 699 3869 7169 9043 155 3848 8229 5033 4555 4558 1878 3046 8663 8106 4418 8517 4314 1853 1767 4990 5464 6170 3354 5589 8471 2454 4768 6780 7105 1593 7956 1085 6803 4520 5
014 1400 6904 3955 8573 7057 7077 4387 1008 9748 2083 1426 5550 5902 754 5561 893 4152 3935 184 3863 2948 7227 6259 6712 6866 3510 8060 5961 6311 2709 225 6218 5133 8249 8170 2181 7651
2120 5333 7445 4237 8445 2698 7822 1634 2701 1518 4322 690 1255 3436 2381 4326 2845 2515 9092 6451 327 9652 914 6395 2548 3759 691 9520 9586 7345 2449 7611 9025 9946 7761 3286 1489 3080
6868 8418 236 6548 9675 6693 3203 1236 956 6962 8861 7248 5655 5276 9801 1285 2565 2173 9054 2048 4609 4021 1900 5335 7720 8568 3610 9926 9280 7202 1392 6142 4778 3485 8239 7532 1320 4
295 5998 4570 8350 6529 7906 7474 814 854 269 7296 3036 447 4413 5461 6985 2202 3859 6152 2826 9794 2362 1466 4741 1728 6204 9122 119 7251 4791 3498 7003 3301 3482 9944 8063 2971 326 82
11 1948 4208 4141 9197 3407 5282 4159 4312 8780 5974 8529 2273 8872 9964 8325 6222 9367 8927 431 5388 9898 230 6384 7230 9232 4086 8768 9565 846 995 461 1290 4053 3278 9587 8785 1530 20
23 1468 6501 3309 7411 6261 2401 5835 2124 4655 4866 1088 4869 7616 5192 6622 9960 4497 9024 8213 2220 8539 6434 8898 9437 1334 9332 5712 3238 825 6273 9883 1274 4230 2770 8244 2869 192
0 3403 6557 2456 6795 7695 8851 3489 9704 4145 5098 8259 4739 1826 1684 9949 7954 5399 8135 5956 5535 3198 1938 2074 1064 511 74 141 6255 8308 1111 1228 9502 4597 2778 8198 8208 6011 81
75 7446 8290 1216 6279 6613 8986 9454 4484 3929 4316 8363 2145 7755 465 9609 6357 6157 8221 5588 8796 5055 4020 7263 1980 4000 8382 7011 9231 5556 321 6669 5750 2815 5504 9787 8807 2326
3112 256 9595 5772 8009 4147 7328 8668 9217 5958 2852 3613 8237 8279 8835 4576 1519 5502 1420 546 1921 3179 9590 8375 9429 6457 8958 3283 8527 8897 7656 1267 1207 401 9860 8109 5190 58
80 4345 9916 6834 7020 4856 2007 1418 3435 7349 6686 6914 1336 9337 1754 6137 5447 9026 1712 9500 6840 4796 5210 1214 8847 8712 2064 2871 5110 9382 1335 3066 5601 7568 187 5940 3513 915
1 5330 8150 6666 1769 486 8528 9789 7683 8912 1456 6323 1779 423 7009 8265 8451 381 4652 3175 6644 5127 7434 4305 5622 1650 9408 4817 9268 7491 1984 985 4339 1221 4267 9049 8871 3989 18
71 6258 719 2989 8667 8422 5841 9365 852 7381 8048 3117 1559 5530 8699 8062 1015 1774 5707 52 6600 5379 1068 5910 6794 521 1538 5401 2759 153 5420 1963 2480 1902 4910 7796 606 4669 1508
3565 9291 1464 7151 4344 654 2121 4483 530 4812 277 3915 9804 6678 6330 9663 6116 2178 7952 9297 5022 722 6951 2344 7834 8326 3748 7887 4061 5491 2348 1680 1325 3492 4008 303 5031 3993
9904 7244 7404 5337 3460 1118 8330 1521 6014 276 8589 7160 60 7275 389 3578 5153 7116 3595 1388 7693 5978 9492 5751 4875 7797 3665 252 504 1376 9677 7877 1933 709 1260 5652 9867 6559 1
674 9320 9464 8031 2556 899 2931 4297 3038 2015 3871 4313 9321 5390 4140 9966 5895 2763 3151 6125 3143 7137 3679 2855 9906 6147 5996 8425 3251 3456 9538 596 2272 4639 2227 2970 1545 836
7 1975 9055 3276 7572 9074 494 3391 7634 970 2859 8085 1549 104 4887 4734 3966 5445 7183 2219 3754 2479 132 1567 5562 7037 9384 5165 7179 2105 8022 6186 9242 3511 2744 919 3185 7973 997
2 1606 8447 3819 9430 8627 4867 4980 5787 53 3379 8041 5862 6362 1845 8552 8973 9307 9123 580 1247 8328 2697 2411 9845 6352 1669 3889 5794 4004 7369 4893 5156 5969 533 5391 2922 3558 21
68 3845 1544 6098 7581 4302 1612 4359 7903 4654 9533 6348 9543 26 4619 884 9350 2031 1561 5804 5783 6419 6862 5685 3965 9773 4263 4054 4915 2571 156 4014 4298 5365 5912 2400 6835 7089 7
387 9357 666 9864 6182 6929 9295 1170 2833 8193 6444 6179 8916 1591 5681 9890 1630 4695 6317 1217 6460 9160 1768 4928 1912 3034 9790 2107 628 2590 8533 8802 1851 7610 9105 2779 7614 261
4 6757 7619 2459 1006 1721 1069 9636 9881 5171 7765 5802 565 6576 6739 4089 1278 4389 5291 9458 4776 6532 1174 9570 6863 4859 7994 8377 3219 6083 6713 3694 5225 4987 5760 3465 1716 5485
1277 7334 5201 1486 8153 781 3699 9798 86 4125 6326 2745 3100 6143 4819 444 377 96 5934 8626 8358 1460 3339 8070 7723 7704 222 17 7365 5687 8950 3324 6735 7413 5411 5090 1510 1223 6550
5724 7388 7727 7025 6019 4607 8762 7503 2643 1613 3499 2391 4157 1145 4522 7870 412 61 5492 674 807 4657 2765 5010 3659 1302 4829 6398 6248 1989 214 1026 3388 2632 2980 3402 6221 4822
4286 7881 9148 1862 7573 6097 5429 528 4575 4181 7280 6199 2967 8311 7519 6239 7689 2642 4035 6191 485 6632 5232 3069 3062 6699 8794 1295 4329 7895 3640 9261 1930 4712 6034 120 8931 982
3055 3639 5437 1189 6542 621 8752 6602 5536 9889 4578 9355 9021 5166 7501 7758 6447 4402 4195 3967 1830 8961 4842 7883 4742 7922 8877 237 5874 9582 5947 2477 2500 6058 5203 6482 3685 2
608 9084 1381 9763 9255 3105 5352 7287 1179 5730 4873 88 7042 1230 3411 9634 7864 3561 3767 7965 299 2856 9637 330 7119 8340 9504 2996 5476 4808 3771 2508 3839 8984 3207 7733 5459 6591
6455 7314 9532 7817 8410 1776 9601 3573 7912 7371 8285 7825 7687 5773 7621 1043 2611 741 2603 4064 2067 8720 4343 4911 5444 7741 9373 5351 761 1639 221 2279 5697 803 2450 7460 9118 9546
5806 7591 4083 8283 6655 2000 2370 9849 7245 963 7957 2143 5656 7850 9718 5785 6949 9524 4981 1925 319 9572 8469 9732 2224 2887 3921 6214 5689 3127 8806 9183 2535 4706 8694 1224 6788 5
983 3693 3583 4033 3326 7041 4045 9472 2648 6737 4745 600 930 2077 1104 6076 8549 2541 1480 446 1101 1681 7425 601 6989 8521 7323 8123 1303 8558 8791 7812 3280 8845 1641 7749 6837 8335
```

Output data - 10000 size random array

* bubble sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582  
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 6  
72 9473 9474 9475 9476 9477 9478 9479 9480 9481 9482 9483 9484 9485 9486 9487 9488 9489 9490 9491 9492 9493 9494 9495 9496 9497 9498 9499 9500 9501 9502 9503 9504 9505 9506 9507 9508 95  
09 9510 9511 9512 9513 9514 9515 9516 9517 9518 9519 9520 9521 9522 9523 9524 9525 9526 9527 9528 9529 9530 9531 9532 9533 9534 9535 9536 9537 9538 9539 9540 9541 9542 9543 9544 9545 95  
46 9547 9548 9549 9550 9551 9552 9553 9554 9555 9556 9557 9558 9559 9560 9561 9562 9563 9564 9565 9566 9567 9568 9569 9570 9571 9572 9573 9574 9575 9576 9577 9578 9579 9580 9581 9582 95  
83 9584 9585 9586 9587 9588 9589 9590 9591 9592 9593 9594 9595 9596 9597 9598 9599 9600 9601 9602 9603 9604 9605 9606 9607 9608 9609 9610 9611 9612 9613 9614 9615 9616 9617 9618 9619 96  
20 9621 9622 9623 9624 9625 9626 9627 9628 9629 9630 9631 9632 9633 9634 9635 9636 9637 9638 9639 9640 9641 9642 9643 9644 9645 9646 9647 9648 9649 9650 9651 9652 9653 9654 9655 9656 96  
57 9658 9659 9660 9661 9662 9663 9664 9665 9666 9667 9668 9669 9670 9671 9672 9673 9674 9675 9676 9677 9678 9679 9680 9681 9682 9683 9684 9685 9686 9687 9688 9689 9690 9691 9692 9693 96  
94 9695 9696 9697 9698 9699 9700 9701 9702 9703 9704 9705 9706 9707 9708 9709 9710 9711 9712 9713 9714 9715 9716 9717 9718 9719 9720 9721 9722 9723 9724 9725 9726 9727 9728 9729 9730 97  
31 9732 9733 9734 9735 9736 9737 9738 9739 9740 9741 9742 9743 9744 9745 9746 9747 9748 9749 9750 9751 9752 9753 9754 9755 9756 9757 9758 9759 9760 9761 9762 9763 9764 9765 9766 9767 97  
68 9769 9770 9771 9772 9773 9774 9775 9776 9777 9778 9780 9781 9782 9783 9784 9785 9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 9800 9801 9802 9803 9804 98  
05 9806 9807 9808 9809 9810 9811 9812 9813 9814 9815 9816 9817 9818 9819 9820 9821 9822 9823 9824 9825 9826 9827 9828 9829 9830 9831 9832 9833 9834 9835 9836 9837 9838 9839 9840 9841 98  
42 9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9854 9855 9856 9857 9858 9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9873 9874 9875 9876 9877 9878 98  
79 9880 9881 9882 9883 9884 9885 9886 9887 9888 9889 9890 9891 9892 9893 9894 9895 9896 9897 9898 9899 9900 9901 9902 9903 9904 9905 9906 9907 9908 9909 9910 9911 9912 9913 9914 9915 99  
16 9917 9918 9919 9920 9921 9922 9923 9924 9925 9926 9927 9928 9929 9930 9931 9932 9933 9934 9935 9936 9937 9938 9939 9940 9941 9942 9943 9944 9945 9946 9947 9948 9949 9950 9951 9952 99  
53 9954 9955 9956 9957 9958 9959 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 9975 9976 9977 9978 9979 9980 9981 9982 9983 9984 9985 9986 9987 9988 9989 99  
90 9991 9992 9993 9994 9995 9996 9997 9998 9999 10000  
10000 size array execution time : 153117 ms
```

Output data - 10000 size random array

* bucket sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
  
83 9584 9585 9586 9587 9588 9589 9590 9591 9592 9593 9594 9595 9596 9597 9598 9599 9600 9601 9602 9603 9604 9605 9606 9607 9608 9609 9610 9611 9612 9613 9614 9615 9616 9617 9618 9619 96  
20 9621 9622 9623 9624 9625 9626 9627 9628 9629 9630 9631 9632 9633 9634 9635 9636 9637 9638 9639 9640 9641 9642 9643 9644 9645 9646 9647 9648 9649 9650 9651 9652 9653 9654 9655 9656 96  
57 9658 9659 9660 9661 9662 9663 9664 9665 9666 9667 9668 9669 9670 9671 9672 9673 9674 9675 9676 9677 9678 9679 9680 9681 9682 9683 9684 9685 9686 9687 9688 9689 9690 9691 9692 9693 96  
94 9695 9696 9697 9698 9699 9700 9701 9702 9703 9704 9705 9706 9707 9708 9709 9710 9711 9712 9713 9714 9715 9716 9717 9718 9719 9720 9721 9722 9723 9724 9725 9726 9727 9728 9729 9730 97  
31 9732 9733 9734 9735 9736 9737 9738 9739 9740 9741 9742 9743 9744 9745 9746 9747 9748 9749 9750 9751 9752 9753 9754 9755 9756 9757 9758 9759 9760 9761 9762 9763 9764 9765 9766 9767 97  
68 9769 9770 9771 9772 9773 9774 9775 9776 9777 9778 9779 9780 9781 9782 9783 9784 9785 9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 9800 9801 9802 9803 9804 98  
05 9806 9807 9808 9809 9810 9811 9812 9813 9814 9815 9816 9817 9818 9819 9820 9821 9822 9823 9824 9825 9826 9827 9828 9829 9830 9831 9832 9833 9834 9835 9836 9837 9838 9839 9840 9841 98  
42 9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9854 9855 9856 9857 9858 9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9873 9874 9875 9876 9877 9878 98  
79 9880 9881 9882 9883 9884 9885 9886 9887 9888 9889 9890 9891 9892 9893 9894 9895 9896 9897 9898 9899 9900 9901 9902 9903 9904 9905 9906 9907 9908 9909 9910 9911 9912 9913 9914 9915 99  
16 9917 9918 9919 9920 9921 9922 9923 9924 9925 9926 9927 9928 9929 9930 9931 9932 9933 9934 9935 9936 9937 9938 9939 9940 9941 9942 9943 9944 9945 9946 9947 9948 9949 9949 9950 9951 9952 99  
53 9954 9955 9956 9957 9958 9959 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 9975 9976 9977 9978 9979 9980 9981 9982 9983 9984 9985 9986 9987 9988 9989 99  
90 9991 9992 9993 9994 9995 9996 9997 9998 9999 10000  
10000 size array execution time : 736 ms
```

Output data - 10000 size random array

* insertion sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536
```

```
57 9658 9659 9660 9661 9662 9663 9664 9665 9666 9667 9668 9669 9670 9671 9672 9673 9674 9675 9676 9677 9678 9679 9680 9681 9682 9683 9684 9685 9686 9687 9688 9689 9690 9691 9692 9693 96  
94 9695 9696 9697 9698 9699 9700 9701 9702 9703 9704 9705 9706 9707 9708 9709 9710 9711 9712 9713 9714 9715 9716 9717 9718 9719 9720 9721 9722 9723 9724 9725 9726 9727 9728 9729 9730 97  
31 9732 9733 9734 9735 9736 9737 9738 9739 9740 9741 9742 9743 9744 9745 9746 9747 9748 9749 9750 9751 9752 9753 9754 9755 9756 9757 9758 9759 9760 9761 9762 9763 9764 9765 9766 9767 97  
68 9769 9770 9771 9772 9773 9774 9775 9776 9777 9778 9779 9780 9781 9782 9783 9784 9785 9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 9800 9801 9802 9803 9804 98  
05 9806 9807 9808 9809 9810 9811 9812 9813 9814 9815 9816 9817 9818 9819 9820 9821 9822 9823 9824 9825 9826 9827 9828 9829 9830 9831 9832 9833 9834 9835 9836 9837 9838 9839 9840 9841 98  
42 9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9854 9855 9856 9857 9858 9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9873 9874 9875 9876 9877 9878 98  
79 9880 9881 9882 9883 9884 9885 9886 9887 9888 9889 9890 9891 9892 9893 9894 9895 9896 9897 9898 9899 9900 9901 9902 9903 9904 9905 9906 9907 9908 9909 9910 9911 9912 9913 9914 9915 99  
16 9917 9918 9919 9920 9921 9922 9923 9924 9925 9926 9927 9928 9929 9930 9931 9932 9933 9934 9935 9936 9937 9938 9939 9940 9941 9942 9943 9944 9945 9946 9947 9948 9949 9950 9951 9952 99  
53 9954 9955 9956 9957 9958 9959 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 9975 9976 9977 9978 9979 9980 9981 9982 9983 9984 9985 9986 9987 9988 9989 99  
90 9991 9992 9993 9994 9995 9996 9997 9998 9999 10000
```

10000 size array execution time : 39343 ms

Output data - 10000 size random array

* merge sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259  
260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305  
306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444  
445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490  
94 9695 9696 9697 9698 9699 9700 9701 9702 9703 9704 9705 9706 9707 9708 9709 9710 9711 9712 9713 9714 9715 9716 9717 9718 9719 9720 9721 9722 9723 9724 9725 9726 9727 9728 9729 9730 9731  
9732 9733 9734 9735 9736 9737 9738 9739 9740 9741 9742 9743 9744 9745 9746 9747 9748 9749 9750 9751 9752 9753 9754 9755 9756 9757 9758 9759 9760 9761 9762 9763 9764 9765 9766 9767 9768  
9769 9770 9771 9772 9773 9774 9775 9776 9777 9778 9779 9780 9781 9782 9783 9784 9785 9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 9800 9801 9802 9803 9804 9805  
9806 9807 9808 9809 9810 9811 9812 9813 9814 9815 9816 9817 9818 9819 9820 9821 9822 9823 9824 9825 9826 9827 9828 9829 9830 9831 9832 9833 9834 9835 9836 9837 9838 9839 9840 9841 9842  
9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9854 9855 9856 9857 9858 9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9873 9874 9875 9876 9877 9878 9879  
9880 9881 9882 9883 9884 9885 9886 9887 9888 9889 9890 9891 9892 9893 9894 9895 9896 9897 9898 9899 9900 9901 9902 9903 9904 9905 9906 9907 9908 9909 9910 9911 9912 9913 9914 9915 9916  
9917 9918 9919 9920 9921 9922 9923 9924 9925 9926 9927 9928 9929 9930 9931 9932 9933 9934 9935 9936 9937 9938 9939 9940 9941 9942 9943 9944 9945 9946 9947 9948 9949 9950 9951 9952 9953  
9954 9955 9956 9957 9958 9959 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 9975 9976 9977 9978 9979 9980 9981 9982 9983 9984 9985 9986 9987 9988 9989 9990  
9991 9992 9993 9994 9995 9996 9997 9998 9999 10000  
10000 size array execution time : 1029 ms
```

Output data - 10000 size random array

* quick sort

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536

```

Output data - 10000 size random array

* radix sort

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65  
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 12  
0 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212  
213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 2  
59 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 30  
5 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351  
352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 4  
44 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 49  
0 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536  
  
57 9658 9659 9660 9661 9662 9663 9664 9665 9666 9667 9668 9669 9670 9671 9672 9673 9674 9675 9676 9677 9678 9679 9680 9681 9682 9683 9684 9685 9686 9687 9688 9689 9690 9691 9692 9693 96  
94 9695 9696 9697 9698 9699 9700 9701 9702 9703 9704 9705 9706 9707 9708 9709 9710 9711 9712 9713 9714 9715 9716 9717 9718 9719 9720 9721 9722 9723 9724 9725 9726 9727 9728 9729 9730 97  
31 9732 9733 9734 9735 9736 9737 9738 9739 9740 9741 9742 9743 9744 9745 9746 9747 9748 9749 9750 9751 9752 9753 9754 9755 9756 9757 9758 9759 9760 9761 9762 9763 9764 9765 9766 9767 97  
68 9769 9770 9771 9772 9773 9774 9775 9776 9777 9778 9779 9780 9781 9782 9783 9784 9785 9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 9800 9801 9802 9803 9804 98  
05 9806 9807 9808 9809 9810 9811 9812 9813 9814 9815 9816 9817 9818 9819 9820 9821 9822 9823 9824 9825 9826 9827 9828 9829 9830 9831 9832 9833 9834 9835 9836 9837 9838 9839 9840 9841 98  
42 9843 9844 9845 9846 9847 9848 9849 9850 9851 9852 9853 9854 9855 9856 9857 9858 9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9873 9874 9875 9876 9877 9878 98  
79 9880 9881 9882 9883 9884 9885 9886 9887 9888 9889 9890 9891 9892 9893 9894 9895 9896 9897 9898 9899 9900 9901 9902 9903 9904 9905 9906 9907 9908 9909 9910 9911 9912 9913 9914 9915 99  
16 9917 9918 9919 9920 9921 9922 9923 9924 9925 9926 9927 9928 9929 9930 9931 9932 9933 9934 9935 9936 9937 9938 9939 9940 9941 9942 9943 9944 9945 9946 9947 9948 9949 9950 9951 9952 99  
53 9954 9955 9956 9957 9958 9959 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 9975 9976 9977 9978 9979 9980 9981 9982 9983 9984 9985 9986 9987 9988 9989 99  
90 9991 9992 9993 9994 9995 9996 9997 9998 9999 10000  
10000 size array execution time : 1009 ms
```

Result

– when input array is random array

Result Table (when random array)

	bubble sort	bucket sort	insertion sort	merge sort	quick sort	radix sort
1000	1113 ms	96 ms	405 ms	92 ms	57 ms	93 ms
5000	33214 ms	371 ms	9806 ms	496 ms	321 ms	417 ms
10000	152736 ms	755 ms	39028 ms	1055 ms	680 ms	1011 ms

```
[~/2022_1/Algorithm/midterm]$ ./program random  
random array
```

* bubble sort

```
1000 size array execution time : 1113 ms  
5000 size array execution time : 33214 ms  
10000 size array execution time : 152736 ms
```

* bucket sort

```
1000 size array execution time : 96 ms  
5000 size array execution time : 371 ms  
10000 size array execution time : 755 ms
```

* insertion sort

```
1000 size array execution time : 405 ms  
5000 size array execution time : 9806 ms  
10000 size array execution time : 39028 ms
```

* merge sort

```
1000 size array execution time : 92 ms  
5000 size array execution time : 496 ms  
10000 size array execution time : 1055 ms
```

* quick sort

```
1000 size array execution time : 57 ms  
5000 size array execution time : 321 ms  
10000 size array execution time : 680 ms
```

* radix sort

```
1000 size array execution time : 93 ms  
5000 size array execution time : 417 ms  
10000 size array execution time : 1011 ms
```

Result

- when input array is reverse array

Result Table (when reverse array)

	bubble sort	bucket sort	insertion sort	merge sort	quick sort	radix sort
1000	2264 ms	72 ms	785 ms	51 ms	736 ms	95 ms
5000	38925 ms	327 ms	19699 ms	290 ms	18239 ms	416 ms
10000	108604 ms	633 ms	78248 ms	580 ms	72310 ms	1029 ms

```
[~/2022_1/Algorithm/midterm]$ ./program reverese  
reverse array
```

* bubble sort

```
1000 size array execution time : 2264 ms  
5000 size array execution time : 38925 ms  
10000 size array execution time : 108604 ms
```

* bucket sort

```
1000 size array execution time : 72 ms  
5000 size array execution time : 327 ms  
10000 size array execution time : 633 ms
```

* insertion sort

```
1000 size array execution time : 785 ms  
5000 size array execution time : 19699 ms  
10000 size array execution time : 78248 ms
```

* merge sort

```
1000 size array execution time : 51 ms  
5000 size array execution time : 290 ms  
10000 size array execution time : 580 ms
```

* quick sort

```
1000 size array execution time : 736 ms  
5000 size array execution time : 18239 ms  
10000 size array execution time : 72310 ms
```

* radix sort

```
1000 size array execution time : 95 ms  
5000 size array execution time : 416 ms  
10000 size array execution time : 1029 ms
```