Lab 31

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  main()
5  {
6      int arr[5][5][5], sortArr[125], i = 0, j, k, n = 5, index = 0, tmp, w_size = 0, h_size = 0, d_size = 0,
7  run = 0, select = 0;
8      char finish = 'y';
9      while (finish == 'y')
10     {
11         while (w_size <= 0 || w_size > 5)
12         {
13             printf("Please input Array Width (maximum = 5): ");
14             scanf("%d", &w_size);
15             if (w_size == 0)
16             {
17                 printf("\n\"Please enter array width  > 0\"\n");
18             }
19             if (w_size > 5)
20             {
21                 printf("\n\"Array Width maximum = 5\"\n\n");
22                 w_size = -1;
23             }
24         }
25         while (h_size <= 0 || h_size > 5)
26         {
27             printf("Please input Array Height size (maximum = 5): ");
28             scanf("%d", &h_size);
```

```
29          if (h_size == 0)
30          {
31              printf("\n\"Please enter height size > 0\"\n");
32          }
33          if (h_size > 5)
34          {
35              printf("\n\"Array Height maximum = 5\"\n\n");
36              h_size = -1;
37          }
38      }
39      while (d_size <= 0 || d_size > 5)
40      {
41          printf("Please input Array Depth (maximum = 5): ");
42          scanf("%d", &d_size);
43          if (d_size == 0)
44          {
45              printf("\n\"Please enter depth  > 0\"\n");
46          }
47          if (d_size > 5)
48          {
49              printf("\n\"Array Depth maximum = 5\"\n\n");
50              d_size = -1;
51          }
52      }
53      while (select != 1 && select != 2)
54      {
55          printf("\n\"Min -> Max(1) or Max -> Min(2)\" (Enter 1 or Enter 2) : ");
56          scanf("%d", &select);
57      }
58
```

```
59          printf("\n");
60          for (i = 0; i < d_size; i++)
61          {
62              for (j = 0; j < h_size; j++)
63              {
64                  for (k = 0; k < w_size; k++)
65                  {
66                      arr[i][j][k] = 0;
67                  }
68              }
69          }
70          for (i = 0; i < w_size * h_size * d_size; i++)
71          {
72              sortArr[i] = 0;
73          }
74          srand(time(NULL));
75          printf("\"Before Sort\"\n");
76          for (i = 0; i < d_size; i++)
77          {
78              for (j = 0; j < h_size; j++)
79              {
80                  for (k = 0; k < w_size; k++)
81                  {
82
83                      arr[i][j][k] = rand() % 200 + 1;
84                      printf("Array[%d][%d][%d] : %d\n", i, j, k, arr[i][j][k]);
85                      sortArr[index] = arr[i][j][k];
86                      index++;
87                  }
88              }
```

```
89          }
90          for(i=0;i<100;i++)
91          {
92                              printf("-");
93                  }
94                  printf("\n\n");
95
96                  printf("\"After Sort\"\n");
97          switch (select)
98          {
99          case 1:
100             for (i = 0; i < w_size * h_size * d_size; i++)
101             {
102                 for (j = 0; j < w_size * h_size * d_size; j++)
103                 {
104                     if (sortArr[j] > sortArr[j + 1])
105                     {
106                         tmp = sortArr[j];
107                         sortArr[j] = sortArr[j + 1];
108                         sortArr[j + 1] = tmp;
109                     }
110                 }
111             }
112             break;
113         case 2:
114             for (i = 0; i < w_size * h_size * d_size; i++)
115             {
116                 for (j = 0; j < w_size * h_size * d_size; j++)
117                 {
118                     if (sortArr[j] < sortArr[j + 1])
```

```
119                 {
120                     tmp = sortArr[j + 1];
121                     sortArr[j + 1] = sortArr[j];
122                     sortArr[j] = tmp;
123                 }
124             }
125         }
126         break;
127     default:
128         break;
129     }
130     index = 0;
131     for (i = 0; i < d_size; i++)
132     {
133         for (j = 0; j < h_size; j++)
134         {
135             for (k = 0; k < w_size; k++)
136             {
137                 arr[i][j][k] = sortArr[index];
138                 index++;
139             }
140         }
141     }
142     for (i = 0; i < d_size; i++)
143     {
144         for (j = 0; j < h_size; j++)
145         {
146             for (k = 0; k < w_size; k++)
147             {
148                 printf("Array[%d][%d][%d] : %d \n", i, j, k, arr[i][j][k]);
```

```
149                 }
150             }
151         }
152         run = 1;
153         while (run == 1)
154         {
155             printf("\nContinue Program ? (y/N) : ");
156             scanf(" %c", &finish);
157             if (finish == 'y' || finish == 'N')
158             {
159                 run = 0;
160                 w_size = 0;
161                 h_size = 0;
162                 d_size = 0;
163                 select = 0;
164                 index = 0;
165             }
166             else
167             {
168                 printf("Enter only \" y \" or \"N\"");
169             }
170         }
171         if (finish == 'N')
172         {
173             printf("\"End Program\"");
174         }
175         printf("\n");
176     }
177 }
```