Lab 35

```
1  #include <stdio.h>
2  #include <time.h>
3
4  void random(int arr[5][5][5], int d_size, int h_size, int w_size);
5  void sort_3d_array(int arr[5][5][5], int d_size, int h_size, int w_size, int select);
6  int arr[5][5][5], sortArr[125], d_size = 0, h_size = 0, w_size = 0;
7  int index = 0, i = 0, j = 0, k = 0, select = 0;
8
9  main()
10 {
11     char finish = 'y', run = 0;
12     while (finish == 'y')
13     {
14         for (i = 0; i < d_size; i++)
15         {
16             for (j = 0; j < h_size; j++)
17             {
18                 for (k = 0; k < w_size; k++)
19                 {
20                     arr[i][j][k] = 0;
21                     sortArr[index] = 0;
22                     index++;
23                 }
24             }
25         }
26         while (w_size <= 0 || w_size > 5)
27         {
28             printf("Please input Array Width (maximum = 5): ");
29             scanf("%d", &w_size);
```

```
30          if (w_size == 0)
31          {
32              printf("\n\"Please enter array width  > 0\"\n");
33          }
34          if (w_size > 5)
35          {
36              printf("\n\"Array Width maximum = 5\"\n\n");
37              w_size = -1;
38          }
39      }
40      while (h_size <= 0 || h_size > 5)
41      {
42          printf("Please input Array Height size (maximum = 5): ");
43          scanf("%d", &h_size);
44          if (h_size == 0)
45          {
46              printf("\n\"Please enter height size > 0\"\n");
47          }
48          if (h_size > 5)
49          {
50              printf("\n\"Array Height maximum = 5\"\n\n");
51              h_size = -1;
52          }
53      }
54      while (d_size <= 0 || d_size > 5)
55      {
56          printf("Please input Array Depth (maximum = 5): ");
57          scanf("%d", &d_size);
58          if (d_size == 0)
59          {
```

```
60              printf("\n\"Please enter depth  > 0\"\n");
61          }
62          if (d_size > 5)
63          {
64              printf("\n\"Array Depth maximum = 5\"\n\n");
65              d_size = -1;
66          }
67      }
68      while (select != 1 && select != 2)
69      {
70          printf("\n\"Min -> Max(1) or Max -> Min(2)\" (Enter 1 or Enter 2) : ");
71          scanf("%d", &select);
72      }
73      printf("\n");
74      random(arr,d_size, h_size, w_size);
75      for(i=0;i<100;i++)
76              {
77                      printf("-");
78              }
79              printf("\n\n");
80
81      sort_3d_array(arr, d_size, h_size, w_size, select);
82      run = 1;
83      while (run == 1)
84      {
85          printf("\nContinue Program ? (y/N) : ");
86          scanf(" %c", &finish);
87          if (finish == 'y' || finish == 'N')
88          {
89              run = 0;
```

```
90                  index = 0;
91                  d_size = 0;
92                  h_size = 0;
93                  w_size = 0;
94                  select = 0;
95                  printf("\n");
96              }
97              else
98              {
99                  printf("Enter only \" y \" or \"N\"");
100             }
101         }
102         if (finish == 'N')
103         {
104             printf("\"End Program\"");
105         }
106     }
107 }
108
109 void random(int arr[5][5][5], int d_size, int h_size, int w_size)
110 {
111         srand(time(NULL));
112         printf("\"Before Sort\"\n");
113         for (i = 0; i < d_size; i++)
114         {
115             for (j = 0; j < h_size; j++)
116             {
117                 for (k = 0; k < w_size; k++)
118                 {
119                     arr[i][j][k] = rand() % 200 + 1;
```

```
120                     printf("Array[%d][%d][%d] : %d\n", i, j, k, arr[i][j][k]);
121                     sortArr[index] = arr[i][j][k];
122                     index++;
123                 }
124             }
125     }
126 }
127
128 void sort_3d_array(int arr[5][5][5], int d_size, int h_size, int w_size, int select)
129 {
130     int tmp = 0;
131     printf("\"After Sort\"\n");
132     switch (select)
133     {
134     case 1:
135         for (i = 0; i < w_size * h_size * d_size; i++)
136         {
137             for (j = 0; j < w_size * h_size * d_size; j++)
138             {
139                 if (sortArr[j] > sortArr[j + 1])
140                 {
141                     tmp = sortArr[j];
142                     sortArr[j] = sortArr[j + 1];
143                     sortArr[j + 1] = tmp;
144                 }
145             }
146         }
147         break;
148     case 2:
149         for (i = 0; i < w_size * h_size * d_size; i++)
```

```
150         {
151             for (j = 0; j < w_size * h_size * d_size; j++)
152             {
153                 if (sortArr[j] < sortArr[j + 1])
154                 {
155                     tmp = sortArr[j + 1];
156                     sortArr[j + 1] = sortArr[j];
157                     sortArr[j] = tmp;
158                 }
159             }
160         }
161         break;
162     default:
163         break;
164     }
165     index = 0;
166     for (i = 0; i < d_size; i++)
167     {
168         for (j = 0; j < h_size; j++)
169         {
170             for (k = 0; k < w_size; k++)
171             {
172                 arr[i][j][k] = sortArr[index];
173                 index++;
174                 printf("Array[%d][%d][%d] : %d \n", i, j, k, arr[i][j][k]);
175             }
176         }
177     }
178 }
```