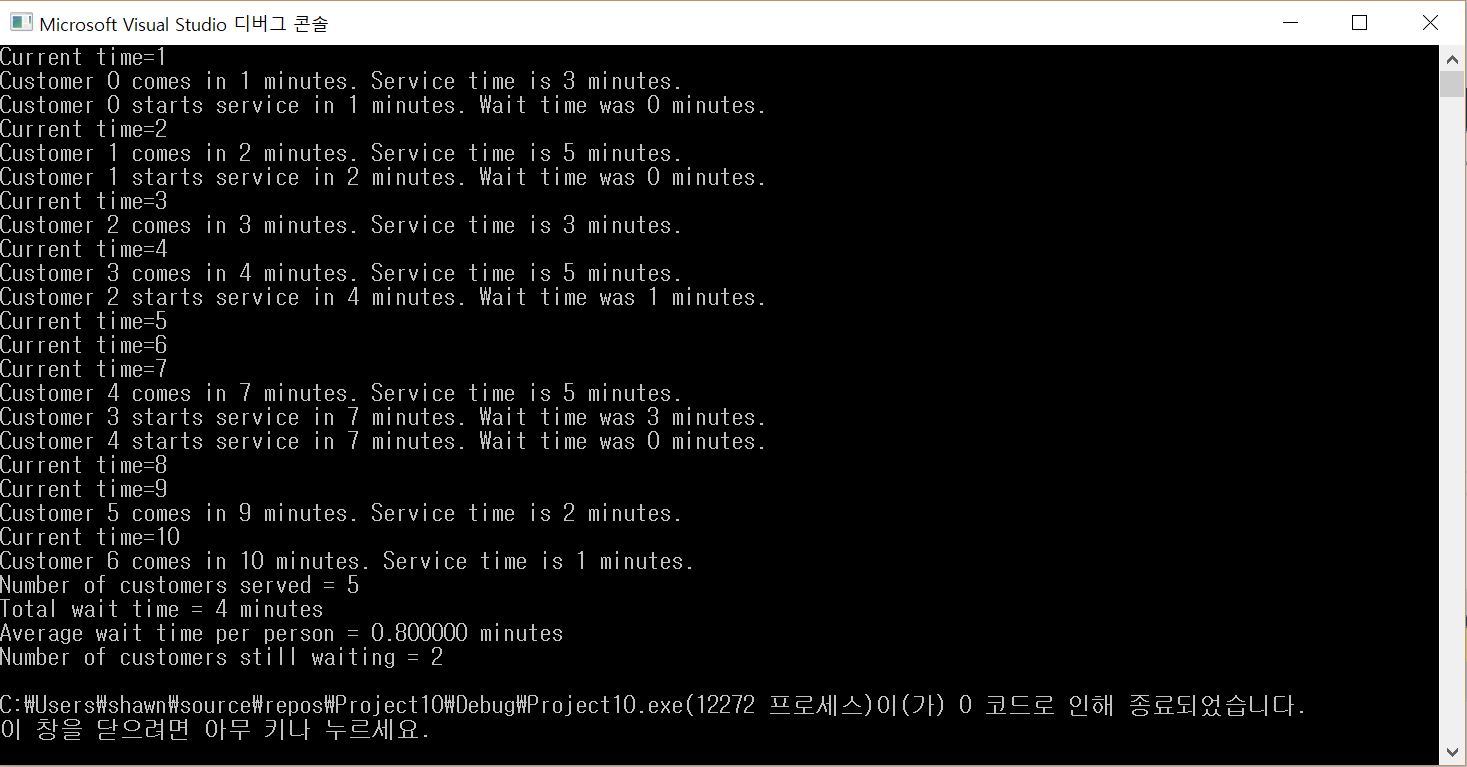
1. 코드 분석 및 주석
2. #include "stdlib.h"
3. #include "stdio.h"
4. #include "string.h"
5. #define MAX\_QUEUE\_SIZE 100
6. typedef struct element {
7. int id;
8. int arrival\_time;
9. int service\_time;
10. } element; // Customer structure
11. typedef struct QueueType {//고객들이 대기하는 장소 queue 구조로 선언
12. element queue[MAX\_QUEUE\_SIZE];
13. int front, rear;
14. } QueueType;
15. QueueType queue;
16. // Real random number generation function between 0 and 1
17. double random() {
18. return rand() / (double)RAND\_MAX;
19. }
20. // Various state variables needed for simulation
21. int duration = 10; // Simulation time
22. double arrival\_prob = 0.7; // Average number of customers arriving in one time unit
23. int max\_serv\_time = 5; // maximum service time for one customer
24. int clock;
25. // Results of the simulation
26. int customers; // Total number of customers
27. int served\_customers; // Number of customers served
28. int waited\_time; // Time the customers waited
29. // Empty state detection function
30. int is\_empty(QueueType \* q)
31. {
32. return (q->front == q->rear);//둘다 null인 경우이니까
33. }
34. // Full state detection function
35. int is\_full(QueueType \* q)
36. {
37. return ((q->rear + 1) % MAX\_QUEUE\_SIZE == q->front);
38. }
39. // Insert function
40. void enqueue(QueueType \* q, element item)
41. {
42. if (is\_full(q))
43. printf("Queue is full\n");
44. q->rear = (q->rear + 1) % MAX\_QUEUE\_SIZE;
45. q->queue[q->rear] = item;
46. }
47. // delete function
48. element dequeue(QueueType \* q)
49. {
50. if (is\_empty(q))
51. printf("Queue is empty\n");
52. q->front = (q->front + 1) % MAX\_QUEUE\_SIZE;
53. return q->queue[q->front];
54. }
55. // Generate a random number.
56. // If it is smaller than ��arrival\_prov��, assume that new customer comes in the bank.
57. int is\_customer\_arrived()
58. {
59. if (random() < arrival\_prob)//random()이 반환하는 숫자가 0.7보다 크다면
60. return true;//고객이 도착한 것으로 간주
61. else return false;
62. }
63. // Insert newly arrived customer into queue
64. void insert\_customer(int arrival\_time)
65. {
66. element customer;
67. customer.id = customers++;//customer 아이디를 하나씩 증가시킴(customers는 전역변수)
68. customer.arrival\_time = arrival\_time;//arrive time을 패러미터로 받아서 element 구조체에 저장
69. customer.service\_time = (int)(max\_serv\_time \* random()) + 1;//random 정수를 받는 방법!! int로 캐스팅 해준다. 그리고 1을 더해줌
70. enqueue(&queue, customer);//대기실에 고객을 집어넣는다.
71. printf("Customer %d comes in %d minutes. Service time is %d minutes.\n", customer.id, customer.arrival\_time, customer.service\_time);
72. }
73. // Retrieve the customer waiting in the queue and return the customer's service time.
74. int remove\_customer()//대기실에서 대기중인 고객을 꺼내서 서비스를 받게 해준다.
75. {
76. element customer;
77. int service\_time = 0;
78. if (is\_empty(&queue)) return 0;
79. customer = dequeue(&queue);//queue에서 빼주기
80. service\_time = customer.service\_time - 1;
81. served\_customers++;
82. waited\_time += clock - customer.arrival\_time;
83. printf("Customer %d starts service in %d minutes. Wait time was %d minutes.\n", customer.id, clock, clock - customer.arrival\_time);
84. return service\_time;
85. }
86. // Print the statistics.
87. void print\_stat()
88. {
89. printf("Number of customers served = %d\n", served\_customers);
90. printf("Total wait time =% d minutes\n", waited\_time);
91. printf("Average wait time per person = %f minutes\n",
92. (double)waited\_time / served\_customers);
93. printf("Number of customers still waiting = %d\n", customers - served\_customers);
94. }
95. // Simulation program
96. void main()
97. {
98. int service\_time1 = 0;//staff가 두명 있으니 서비스 타임 2개 선언
99. int service\_time2 = 0;
100. clock = 0;
101. while (clock < duration) {//duration안에서 수행
102. clock++;
103. printf("Current time=%d\n", clock);
104. if (is\_customer\_arrived()) {//customer가 도착했다면
105. insert\_customer(clock);//insert 해준다. clock 시간과 함께
106. }
107. // Check if the customer who is receiving the service is finished.
108. if (service\_time1 > 0) // the customer is receiving service
109. service\_time1--;
110. else {
111. service\_time1 = remove\_customer();
112. }
113. if (service\_time2 > 0)//2번째 staff의 시간도 체크해준다.
114. service\_time2--;
115. else {//2번째 staff가 비어있다면
116. service\_time2 = remove\_customer();//대기실에서 꺼내서 서비스를 받게 해준다.
117. }
118. }
119. print\_stat();//출력
120. }

2. 결과콘솔창



3. 주요 코드 분석

1. 2명의 staff를 위한 두개의 시간변수 선언
2. int service\_time1 = 0;//staff가 두명 있으니 서비스 타임 2개 선언
3. int service\_time2 = 0;

2) 1번 staff 앞이 비어있지 않으면 2번 staff의 앞이 비어있는지 순차적으로 확인하는 코드

1. if (service\_time1 > 0) // the customer is receiving service
2. service\_time1--;
3. else {
4. service\_time1 = remove\_customer();
5. }
6. if (service\_time2 > 0)//2번째 staff의 시간도 체크해준다.
7. service\_time2--;
8. else {//2번째 staff가 비어있다면
9. service\_time2 = remove\_customer();//대기실에서 꺼내서 서비스를 받게 해준다.