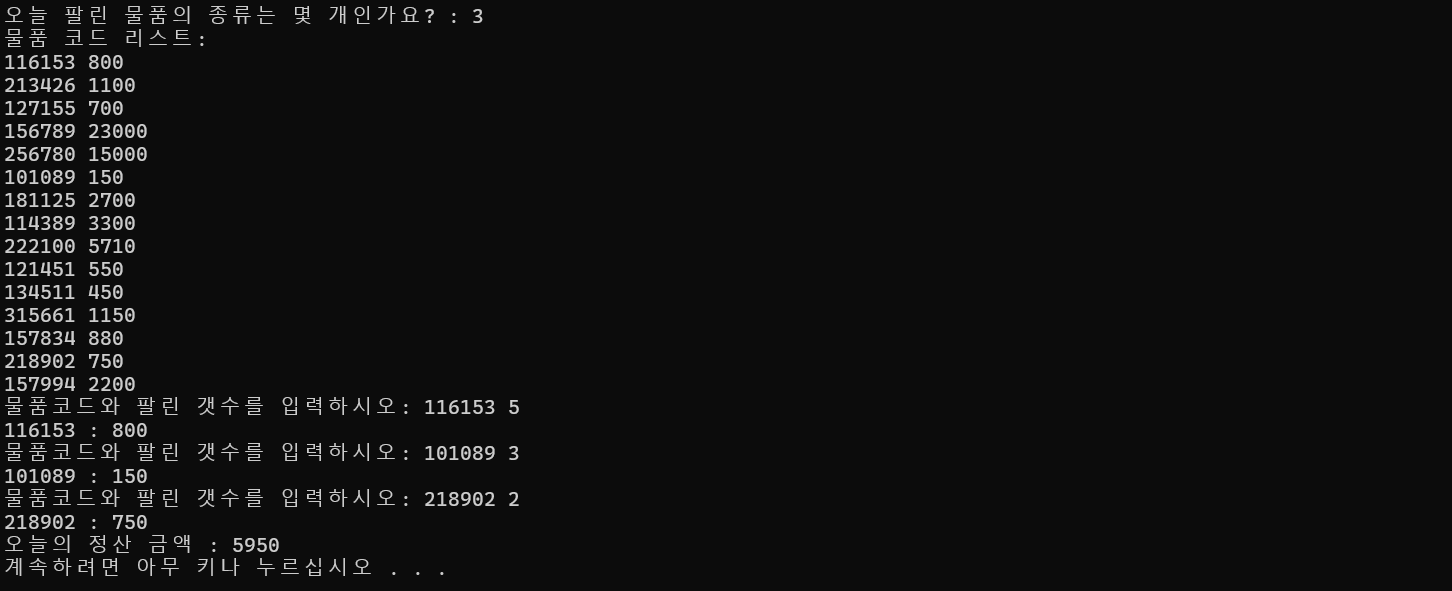
**알고리즘YD 20212127 송하성**

**9주차 과제**

**룩업 테이블 만들기 )**

|  |
| --- |
| C |
| #define \_CRT\_SECURE\_NO\_WARNINGS //355  #include <stdio.h>  #define MAX\_NUM 500  long lookup(long pdb[][2], int n, long item\_code) {  int i;  for (i = 0; i < n; i++) {  if (pdb[i][0] == item\_code) return pdb[i][1];  }  return 0;  }  /\*long stocksum(unsigned s[][2], int n) {  long sum = 0;  int i;  for (i = 0; i < n; i++)  sum += s[i][1];  return sum;  }\*/  void display\_code(long pdb[][2], int n) {  int i;  printf("물품 코드 리스트: \n");  for (i = 0; i < n; i++)  printf("%ld\n", pdb[i][0]);  }  void main(int argc, char\* argv[]) {  FILE\* pricedb;  long pdb[MAX\_NUM][2], item\_code, item\_price;  int i = 0, num, n = 0, item\_num;  long total\_price = 0;  if ((pricedb = fopen(argv[1], "r")) == NULL) {  printf("데이터 파일을 열 수 없습니다.\n");  exit(1);  }  while (fscanf(pricedb, "%ld %ld", &(pdb[i][0]), &(pdb[i][1])) != EOF)  i++;  n = i;    printf("오늘 팔린 물품의 종류는 몇 개인가요? : ");  scanf("%d", &num);  display\_code(pdb, n);  for (int j = 0; j < num; j++) {  printf("물품코드와 팔린 갯수를 입력하시오: ");  scanf("%ld %ld", &item\_code, &item\_num);  item\_price = lookup(pdb, n, item\_code);  if (item\_price == 0) {  printf("%ld : 입력한 물품에 대한 가격 정보가 없습니다", item\_code);  }  else {  printf("%ld : %ld\n", item\_code, item\_price);  }  total\_price = total\_price + (item\_price \* item\_num);  }  printf("오늘의 정산 금액 : %ld\n", total\_price);  system("pause");  } |



스택을 이용한 알고리즘 ) 결과값 3개 출력

|  |
| --- |
| C |
| #define \_CRT\_SECURE\_NO\_WARNINGS //p.218  #include <stdio.h>  #define MAX\_EXPR\_SIZE 50  #define MAX\_STACK\_SIZE 30  char pexpr[MAX\_EXPR\_SIZE];  int stack[MAX\_STACK\_SIZE];  int top=-1;  int delete\_stack();  void add\_stack(int data);  int is\_operator(char c);  int cal(void);  void main() {  printf("Input the expression as a postfix notatin : \n");  gets(pexpr);  printf("%s", pexpr);  printf("Evaluation Value : %d\n", cal());  }  int delete\_stack() {  int data;  if (top == -1) {  printf("Stack is empty. \n");  exit(1);  }  data = stack[top]; top--; return data;  //return stack[top--]  }  void add\_stack(int data) {  if (top >= MAX\_STACK\_SIZE-1) {  printf("Stack is full. \n");  exit(1);  }  ++top; stack[top] = data;  //stack[++top] = data;  }  int is\_operator(char c) {  if (c == '+' || c == '-' || c == '\*' || c == '/')  return 1;  else  return 0;  }  int cal(void) {  char symbol;  int op1, op2, n = 0;  symbol = pexpr[n++];  while (symbol != '\0') {  if (is\_operator(symbol)) {  //피연산자를 삭제하여 연산을 수행한 후 그 결과를 스택에 삽입  op2 = delete\_stack();  op1 = delete\_stack();  printf("%d %d", op1, op2);  switch (symbol) {  case '+':add\_stack(op1 + op2);  break;  case '-':add\_stack(op1 - op2);  break;  case '\*':add\_stack(op1 \* op2);  break;  case '/':add\_stack(op1 / op2);  break;  }  }  else  add\_stack(symbol - '0'); //스택에 삽입  symbol = pexpr[n++];  }  return delete\_stack();  } |





