문제 분석)

백화점에서 물건을 사는 종류와 개수에 따라 경품 응모 번호를 받고 당첨 번호와 비교하여 일치하는 번호의 갯수에 따라 등수를 출력한다.

구매자는 20만원의 TV, 10만원의 냉장고, 5만원의 청소기, 만원의 오븐을 살 수 있으며, 각각 10번, 5번, 2번, 1번의 경품 응모 기회를 얻는다.

응모 번호는 중복되지 않고 랜덤하게 생성해야하며, 물품 구매 후 누적된다.  
당첨 번호는 1~100 사이의 중복되지 않는 랜덤한 번호 10개이고, 같은 번호가 10개이면 1등, 8~9개이면 2등, 5~7개이면 3등, 2~4개이면 4등, 0~1개이면 5등이다.

소지할 수 있는 초기 금액은 0~500만원 사이이다.

의사 코드 및 소스 코드 분석)

check\_duplicate(arr[], size, num) {

for i = 0; i < size; i++{

if num == arr[i] then

return 1

return 0

}

get\_match(answer[], data[], size) {

set match to 0;

for i = 0; i < 10; i++ {

for j = 0; j < size; j++ {

if answer[i] == data[j] then

match++

break

}

}

return match

}

main() {

print "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

print "<중간프로젝트2>\n"

print "학번 : 2020XXXXX, 이름 : 홍길동\n\n"

print "백화점에서 물건을 구매하십시오.\n\n"

print "TV의 가격은 20만원이며 응모 기회는 10번 주어집니다.\n"

print "냉장고의 가격은 10만원이며 응모 기회는 5번 주어집니다.\n"

print "청소기의 가격은 5만원이며 응모 기회는 2번 주어집니다.\n"

print "오븐의 가격은 1만원이며 응모 기회는 1번 주어집니다.\n\n"

set budget

set chance to 0

print "당신의 현재 보유 자금은?(x만원) : “

get budget

while budget > 500 or budget < 0 {

print "잘못된 금액입니다. 0~500만원 이하의 금액을 입력하세요.\n\n"

print "당신의 현재 보유 자금은?(x만원) : "

get budget

}

set shopping to 1

while shopping {

set product, amount

print "\n무엇을 구매하시겠습니까?\n"

print "1.TV 2.냉장고 3.청소기 4.오븐 5.구입을 멈춘다\n" // "물건 구입 메뉴"

get product

switch (product) {

case 1:

print "\n내가 구매할 TV의 개수는? : "

get amount

if budget < amount \* 20 then

print "보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 20 - budget

if budget < 20 then

print "현재 잔돈은 %d만원이며, TV는 더 이상 구매할 수 없습니다.\n", budget

else then

print "현재 잔돈은 %d만원이며, TV는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 20

else then

budget -= amount \* 20;

chance += amount \* 10;

print "잔돈은 %d만원입니다\n", budget

break;

case 2:

print "\n내가 구매할 냉장고의 개수는? : "

get amount

if budget < amount \* 10 then

print "보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 10 - budget

if budget < 10 then

print "현재 잔돈은 %d만원이며, 냉장고는 더 이상 구매할 수 없습니다.\n", budget

else then

print "현재 잔돈은 %d만원이며, 냉장고는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 10

else then

budget -= amount \* 10;

chance += amount \* 5;

print "잔돈은 %d만원입니다\n", budget

break;

case 3:

print "\n내가 구매할 청소기의 개수는? : "

get amount

if budget < amount \* 5 then

print "보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 5 - budget

if budget < 5

print "현재 잔돈은 %d만원이며, 청소기는 더 이상 구매할 수 없습니다.\n", budget

else then

print "현재 잔돈은 %d만원이며, 청소기는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 5

else then

budget -= amount \* 5;

chance += amount \* 2;

print "잔돈은 %d만원입니다\n", budget

break;

case 4:

print "\n내가 구매할 오븐의 개수는? : "

get amount

if budget < amount \* 1 then

print "보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 1 - budget

if budget < 1 then

print "현재 잔돈은 %d만원이며, 오븐은 더 이상 구매할 수 없습니다.\n", budget

else then

print "현재 잔돈은 %d만원이며, 오븐은 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 1

else then

budget -= amount \* 1;

chance += amount \* 1;

print "잔돈은 %d만원입니다\n", budget);

break;

case 5:

set shopping to 0;

break;

default:

print "잘못된 선택입니다. 다시 선택해주세요.\n"

break;

}

}

print "\n더 이상 물건을 사지 않습니다\n"

print "경품 응모를 시작합니다\n\n\n"

print "내 응모 기호는 : %d\n", chance

set chances[100]

print "경품 응모 번호는 :"

for i = 0; i < chance; i++ {

set num to rand % 100 + 1;

if i >= 100 then

break;

if check\_duplicate(chances, i, num) then

i--;

continue;

else then

chances[i] = num;

print " %d", chances[i]);

}

print "\n"

set answer[10]

print "경품 당첨 번호는 : "

for i = 0; i < 10; i++ {

set num to rand % 100 + 1;

if check\_duplicate(answer, i, num) then

i--;

continue;

else then

answer[i] = num;

print " %d", answer[i]);

}

print "\n"

int match = get\_match(answer, chances, chance);

print "일치하는 번호 개수: %d\n", match);

set rank

if match >= 10 then

set rank to 1

else if match >= 8 then

set rank to 2

else if match >= 5 then

set rank to 3

else if match >= 1 then

set rank to 4

else

set rank to 5

print "당신의 등수: %d등\n", rank

return 0

}

———————————————————————————————————————

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

// 중복 체크 함수

int check\_duplicate(int arr[], int size, int num)

{

// 배열의 크기만큼 순회하며, num과 일치하는 숫자가 있으면 1, 없으면 0을 반환

for (int i = 0; i < size; i++)

if (num == arr[i])

return 1;

return 0;

}

// 일치하는 응모 번호와 당첨 번호의 갯수를 반환

int get\_match(int answer[], int data[], int size) {

// 당첨 갯수를 0으로 초기화

int match = 0;

// 10개의 당첨 번호에 대해서

for (int i = 0; i < 10; i++)

// 모든 응모 번호에 대해 검사

for (int j = 0; j < size; j++) {

// 일치하면 당첨 갯수 증가

if (answer[i] == data[j]) {

match++;

break;

}

}

// 당첨 갯수 반환

return match;

}

// 메인 함수

int main() {

// 개인 정보 및 프로젝트 내용 기본 출력

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("<중간프로젝트2>\n");

printf("학번 : 2020XXXXX, 이름 : 홍길동\n\n");

printf("백화점에서 물건을 구매하십시오.\n\n");

printf("TV의 가격은 20만원이며 응모 기회는 10번 주어집니다.\n");

printf("냉장고의 가격은 10만원이며 응모 기회는 5번 주어집니다.\n");

printf("청소기의 가격은 5만원이며 응모 기회는 2번 주어집니다.\n");

printf("오븐의 가격은 1만원이며 응모 기회는 1번 주어집니다.\n\n");

// 예산과 응모 기회를 0으로 초기화 후 예산을 입력받는다.

int budget = 0, chance = 0;

printf("당신의 현재 보유 자금은?(x만원) : "); // 프로그램 안내 메시지

scanf(" %d", &budget);

// 예산의 범위가 잘못 되었으면 올바른 입력이 들어올때까지 반복

while (budget > 500 || budget < 0) {

printf("잘못된 금액입니다. 0~500만원 이하의 금액을 입력하세요.\n\n");

printf("당신의 현재 보유 자금은?(x만원) : ");

scanf(" %d", &budget);

}

// 쇼핑을 계속 하고있는지에 대한 상태를 저장할 변수

int shopping = 1;

// 쇼핑중인 동안 반복

while (shopping) {

// 상품과 수량을 저장할 변수

int product, amount;

printf("\n무엇을 구매하시겠습니까?\n");

printf("1.TV 2.냉장고 3.청소기 4.오븐 5.구입을 멈춘다\n"); // "물건 구입 메뉴”

// 상품을 먼저 입력받음.

scanf(" %d", &product);

switch (product) {

// 1~4 메뉴 선택 시, 각 물건을 A라 하면 "내가 구매할 A의 개수는? : " 출력 이후 "잔돈은 n만원입니다" 출력

// 잔돈을 초과하여 구매 시 "보유한 자금을 %d만원만큼 초과하였습니다." 출력

// 이후 해당 물건의 구매 가능 개수가 0개이면 "현재 잔돈은 n만원이며, A는 더 이상 구매할 수 없습니다." 출력, 구매가능 개수가 x개이면 "현재 잔돈은 n만원이며, A는 x개 이하로 구매할 수 있습니다." 출력 이후 다시 "물건 구입 메뉴" 출력

case 1:

// 갯수를 입력 받음

printf("\n내가 구매할 TV의 개수는? : ");

scanf("%d", &amount);

// 예산이 구매할 상품의 수량보다 적을 경우

if (budget < amount \* 20) {

printf("보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 20 - budget);

// 하나의 제품도 구매할 수 없는 경우

if (budget < 20)

printf("현재 잔돈은 %d만원이며, TV는 더 이상 구매할 수 없습니다.\n", budget);

// 하나 이상 구매가능한 경우

else

printf("현재 잔돈은 %d만원이며, TV는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 20);

// 입력값 만큼 구매가능한 경우

} else {

// 예산을 가격 \* 수량 만큼 줄이고, 기회 \*수량만큼 기회 증가

budget -= amount \* 20;

chance += amount \* 10;

printf("잔돈은 %d만원입니다\n", budget);

}

break;

case 2:

// 갯수를 입력 받음

printf("\n내가 구매할 냉장고의 개수는? : ");

scanf("%d", &amount);

// 예산이 구매할 상품의 수량보다 적을 경우

if (budget < amount \* 10) {

printf("보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 10 - budget);

// 하나의 제품도 구매할 수 없는 경우

if (budget < 10)

printf("현재 잔돈은 %d만원이며, 냉장고는 더 이상 구매할 수 없습니다.\n", budget);

// 하나 이상 구매가능한 경우

else

printf("현재 잔돈은 %d만원이며, 냉장고는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 10);

// 입력값 만큼 구매가능한 경우

} else {

// 예산을 가격 \* 수량 만큼 줄이고, 기회 \*수량만큼 기회 증가

budget -= amount \* 10;

chance += amount \* 5;

printf("잔돈은 %d만원입니다\n", budget);

}

break;

case 3:

// 갯수를 입력 받음

printf("\n내가 구매할 청소기의 개수는? : ");

scanf("%d", &amount);

// 예산이 구매할 상품의 수량보다 적을 경우

if (budget < amount \* 5) {

printf("보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 5 - budget);

// 하나의 제품도 구매할 수 없는 경우

if (budget < 5)

printf("현재 잔돈은 %d만원이며, 청소기는 더 이상 구매할 수 없습니다.\n", budget);

// 하나 이상 구매가능한 경우

else

printf("현재 잔돈은 %d만원이며, 청소기는 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 5);

// 입력값 만큼 구매가능한 경우

} else {

// 예산을 가격 \* 수량 만큼 줄이고, 기회 \*수량만큼 기회 증가

budget -= amount \* 5;

chance += amount \* 2;

printf("잔돈은 %d만원입니다\n", budget);

}

break;

case 4:

// 갯수를 입력 받음

printf("\n내가 구매할 오븐의 개수는? : ");

scanf("%d", &amount);

// 예산이 구매할 상품의 수량보다 적을 경우

if (budget < amount \* 1) {

printf("보유한 자금을 %d만원만큼 초과하였습니다.\n", amount \* 1 - budget);

// 하나의 제품도 구매할 수 없는 경우

if (budget < 1)

printf("현재 잔돈은 %d만원이며, 오븐은 더 이상 구매할 수 없습니다.\n", budget);

// 하나 이상 구매가능한 경우

else

printf("현재 잔돈은 %d만원이며, 오븐은 %d개 이하로 구매할 수 있습니다.\n", budget, budget / 1);

// 예산을 가격 \* 수량 만큼 줄이고, 기회 \*수량만큼 기회 증가

} else {

budget -= amount \* 1;

chance += amount \* 1;

printf("잔돈은 %d만원입니다\n", budget);

}

break;

case 5:

// 5.구입을 멈춘다 선택 시 "더 이상 물건을 사지 않습니다", "경품 응모를 시작합니다" 출력 이후 "경품 응모 단계"로 넘어감

// 쇼핑 상태를 하지 않는 상태로 변경

shopping = 0;

break;

default:

// 1~5 외의 메뉴 선택 시 "잘못된 선택입니다. 다시 선택해주세요" 출력 후 다시 "물건 구입 메뉴" 출력

printf("잘못된 선택입니다. 다시 선택해주세요.\n");

break;

}

}

// "경품 응모 단계"

printf("\n더 이상 물건을 사지 않습니다\n");

printf("경품 응모를 시작합니다\n\n\n");

// "내 응모 기회는 : %d" 출력

printf("내 응모 기호는 : %d\n", chance);

// 예산이 최대일 경우, TV를 25개, 즉 250번의 응모 기회를 최대로 가질 수 있다. 하지만 문제 조건에서 최대 100개의 난수를 생성하라고 명시되어 있으므로, 크기가 100인 배열로 생성.

int chances[100];

printf("경품 응모 번호는 :"); // 중복되지 않는 랜덤 경품 응모 번호 출력

// 총 응모 기회만큼 반복

for (int i = 0; i < chance; i++) {

// 1 ~ 100 사이의 난수 생성

int num = rand() % 100 + 1;

// 100회가 넘어갈 경우 반복문 종료

if (i >= 100)

break;

// 중복된 수가 생성되었는지 검사하고, 있으면 현재 인덱스에서 뒤로 한칸 이동하여 새로운 난수로 덮어씀

if (check\_duplicate(chances, i, num)) {

i--;

continue;

} else {

// 중복이 없으면 생성된 수 저장 후 출력

chances[i] = num;

printf(" %d", chances[i]);

}

}

printf("\n");

// 당첨 번호는 총 10개이다.

int answer[10];

printf("경품 당첨 번호는 : "); // 중복되지 않는 랜덤 경품 당첨 번호 출력

// 응모 기회와 같은 방법으로 수행

for (int i = 0; i < 10; i++) {

int num = rand() % 100 + 1;

if (check\_duplicate(answer, i, num)) {

i--;

continue;

} else {

answer[i] = num;

printf(" %d", answer[i]);

}

}

printf("\n");

// "일치하는 번호 개수: %d개" 출력

int match = get\_match(answer, chances, chance);

printf("일치하는 번호 개수: %d\n", match);

// "당신의 등수: %d등" 출력

int rank;

// 10개 이상일 경우

if (match >= 10)

rank = 1;

// 8개 이상일 경우

else if (match >= 8)

rank = 2;

// 5개 이상일 경우

else if (match >= 5)

rank = 3;

// 2개 이상일 경우

else if (match >= 2)

rank = 4;

// 이외의 경우

else

rank = 5;

printf("당신의 등수: %d등\n", rank);

return 0;

}

int check\_duplicate(int arr[], int size, int num)

주어진 배열 내에 num 매개변수와 같은 값을 가진 변수값이 존재하는지 검사 후 0(거짓) 또는 1(참)을 반환하는 함수이다. arr 배열을 0부터 size의 크기만큼 돌면서 검사하며, 만약 num과 같은 값을 가진 배열 변수값이 존재할 경우 그 즉시 참을 반환한다.

int get\_match(int answer[], int data[], int size)

answer 배열과 data 배열을 이용하여 경품 추첨 결과로 몇개의 번호가 일치하는지 반환하는 함수이다. answer 배열은 항상 길이가 10인 정답 배열을 매개 변수로 받고, 그 각각의 배열 변수값에 대해서 data 배열을 size 만큼 0부터 순회하며, 해당 반복중인 answer 변수 값과 일치하는 data 변수 값이 있으면 정답갯수를 1 증가시킨 후, 반복문을 즉시 종료한다. 모든 정답 배열 순회가 끝난 후에 같은 변수의 갯수를 반환한다.

int main()

프로그램의 메인 함수이다. 프로그램 실행 시 주어진 출력문들을 출력하고, 맨 처음 자금을 입력받으며, 자금의 조건에 일치할 때까지 계속해서 입력받는다. 그 이후, 사용자의 쇼핑 상태를 나타내는 병수를 이용하여 구입을 멈출 때 까지 반복해서 구매할 상품의 번호를 올바은 입력을 받을 때까지 반복해서 입력받고, 구매할 상품의 갯수를 입력받아 자금 상황과 총 구매 금액을 비교하여 자금이 충분하다면 구매와 동시에 응모 기회를 증가시키고, 그렇지 못하면 그에 알맞는 출력을 진행한다. 구매가 끝나고 나면, 사용자의 총 응모 기회를 출력하고, 0부터 응모 기회만큼 반복문을 통해 경품 응모 번호를 생성하는데, 반복 횟수는 최대 100회로써 응모기회만큼 반복한다. 매 반복마다 난수를 생성하여, check\_duplicate 함수를 호출하여 생성된 난수가 배열안에 존재하는지 검사하고, 중복하지 않는다면 배열에 저장 및 출력한다. 같은 방식으로 당첨번호를 10회 반복하여 생산하고, get\_match 함수를 통해 일치하는 번호의 갯수를 생성, 사용자의 응모 숫자와 비교하여 등수를 출력하며 프로그램을 종료한다.

결과)