|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5주차 과제 | | | | | |
| 학년 | 2 | 학번 | 2016707079 | 이름 | 하상천 |

1. 과제 설명

과제1-1

문자의 개수가 5인 문자열을 입력 받고 key 값을 입력해서 암호화 하는 코드를 작성 한다.

Key 값은 1~26이 되도록 한다. 문자열에 key값을 각각 더해서 알파벳이 바뀌도록 한다.

(예시 a에 key 값 1을 더하면 b가 된다.) z에서 숫자를 더할 경우 다시 a부터 돌아가 순환 되도록 한다. 문자열은 알파벳 소문자로만 이루어 지게 입력한다.

과제 1-2

1-1에서 만든 암호화 된 문자열에서 key값을 빼서 복호화 하는 코드를 작성한다.

Key 값은 1~26이 되도록 한다. a에서 숫자를 뺄 경우 반대로 z부터 돌아가 순환 되도록 한다.(1-1과는 반대로 작동 하게 설정)

과제 1-3

암호화 된 문자열에서 key값을 모를 경우 전수조사라는 방식을 통해서 복호화를 한다.

Key 값이 될 수 있는 1과 26사이의 모든 숫자들을 이용해서 복호화를 진행 한다.(복호화 총 26번 반복) 전수조사 한 결과 중에서 원래 암호를 추정한다.

과제 2

길이가 15인 배열 A, B를 조합하여 길이가 30인 배열 C를 만든다.

배열 A,B는 길이가 15인 char형 배열로 선언 및 위의 값으로 초기화한다.

배열 C는 길이가 30인 char형 배열로 선언한다.

배열 C의 값은 규칙에 따라 개수를 증가시켜가며 입력을 한다.

****

1. 과제 이론

If문

조건을 주어 그 조건을 만족할 때에 실행하도록 한다.

Ex) if(i==1) 이면 i의 값이 1일 때만 실행하도록 한다.

그 외에는 다른 실행을 하고 싶다면 똑같은 형식으로 else if 또는 else로 조건을 주면 된다.

For문

일정한 조건을 주어서 원하는 만큼 반복할 수 있게 해주는 반복문

Ex) for(i=0;i<10;i++); ---> i=0으로 시작하고 한번 반복 할 때마다 i값이 1씩 증가한다.

그리고 i<10이라는 조건이 만족 할 때까지만 반복한다. 즉 i=10이되면 반복하지 않는다.

Do…while문

Do{

반복할 코드(적어도 한번 실행)

}while(종결조건);

1. 주요 소스 설명

과제1

#include<stdio.h>

int main()

{

char name[6], newname[6];

int key, i;

printf("문자열을 입력하세요\n");

scanf("%s", &name); //문자열 입력받기

do {

printf("key값을 입력하세요\n");

scanf("%d", &key); //key값 입력받기

} while (key < 1 || key>26); //key값으로 1부터 26사이의 값을 입력하지 않으면 다시 반복한다.

for (i = 0; i < 5; i++)

{

newname[i] = name[i] + key; //각각의 문자열에 key값을 더한다.

}

for (i = 0; i < 5; i++)

{

if (newname[i] > 122) //알파벳 z에 숫자를 더했을 때 다시 a부터 순환하도록 한다. (ASCII 코드 참고)

{

newname[i] -= 26;

}

}

printf("기본 문자열 %s\n암호화 된 문자열 ", name);

for (i = 0; i < 5; i++) //암호화 된 문자열 출력

{

printf("%c", newname[i]);

}

printf("\n");

do {

printf("key값을 입력하세요\n");

scanf("%d", &key); //key값 입력받기

} while (key < 1 || key>26); //key값으로 1부터 26사이의 값을 입력하지 않으면 다시 반복한다.

for (i = 0; i < 5; i++) // 암호화 된 문자열에서 key 값을 뺀다.

{

name[i] = newname[i] - key;

}

for (i = 0; i < 5; i++)

{

if (name[i] < 97) //알파벳 a에서 숫자를 뺄 경우 반대로 z부터 순환하도록 한다. (ASCII 코드 참고)

{

name[i] += 26;

}

}

printf("복호화 된 문자열 ");

for (i = 0; i < 5; i++) //복호화 된 문자열을 출력한다.

{

printf("%c", name[i]);

}

printf("\n암호화 된 문자열 ");

for (i = 0; i < 5; i++) //암호화 된 문자열을 출력한다.

{

printf("%c", newname[i]);

}

for (i = 0; i < 27; i++) // 1과 26사이의 모든 숫자들을 이용해서 복호화를 진행 한다.

{

printf("\nkey 값 %d 복호화 된 문자열 ", i);

for (int j = 0; j < 5; j++)

{

if (newname[j]-i < 97) //알파벳 a에서 숫자를 뺄 경우 반대로 z부터 순환하도록 한다.

{

printf("%c", newname[j] - i +26);

}

else

{

printf("%c", newname[j] - i);

}

}

}

}

과제 2

#include<stdio.h>

int main()

{

char A[15] = { '0','1','2','3','4','5','6','7','8','9','A','B','C','D','E' }; //배열 A를 선언하고 주어진 값으로 초기화

char B[15] = { 'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T' };//배열 B를 선언하고 주어진 값으로 초기화

char C[30]; //배열C 선언

int index\_a = 0; //배열 A의 출발점

int index\_b = 14; //배열 B의 출발점

int up = 1;

int down = 5;

int k = 0;

for (int x = 0; x < 5; x++) //A배열, B배열에서 순차적으로 입력하는 것을 5번 반복한다.

{

for (int i = index\_a; i < index\_a + up; i++) //A배열의 값을 C배열의 값으로 입력하는 반복문

{

C[k] = A[i];

k++;

}

index\_a += up; // 변수 index\_a에 변수up을 더해서 다음 반복문에 초기값을 맞추어준다.

up++;

for (int j = index\_b; j > index\_b - down; j--) //B배열의 값을 C배열의 값으로 입력하는 반복문

{

C[k] = B[j];

k++;

}

index\_b -= down; // 변수 index\_b에 변수down을 빼서 다음 반복문에 초기값을 맞추어준다.

down--;

}

printf("A배열 : ");

for (int x = 0; x < 15; x++) //A배열 출력 , %2c는 한칸씩 띄어서 쓰기 위해서이다.

{

printf("%2c", A[x]);

}

printf("\nB배열 : ");

for (int x = 0; x < 15; x++)//B배열 출력 , %2c는 한칸씩 띄어서 쓰기 위해서이다.

{

printf("%2c", B[x]);

}

printf("\nC배열 : "); //C배열 출력 , %2c는 한칸씩 띄어서 쓰기 위해서이다.

for (int x = 0; x < 30; x++)

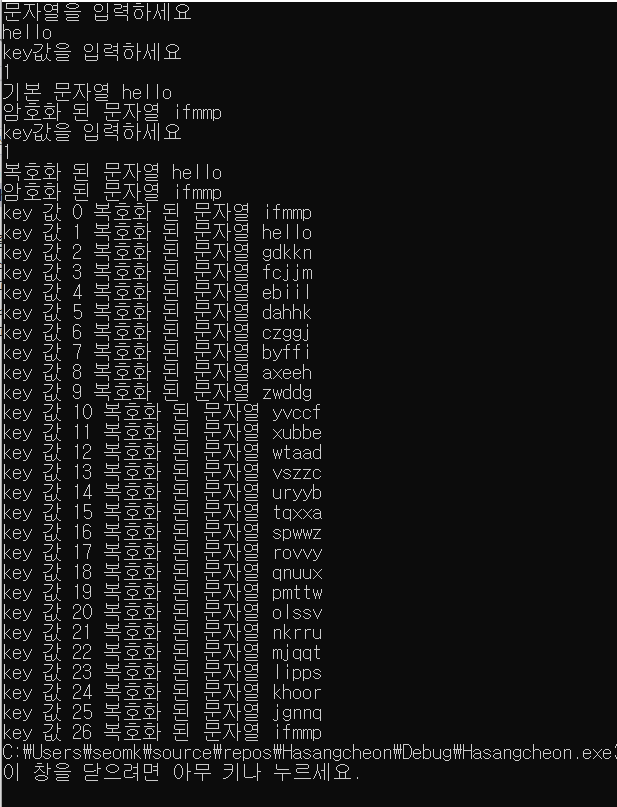
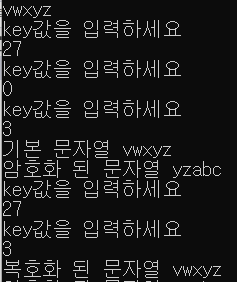
{

printf("%2c", C[x]);

}

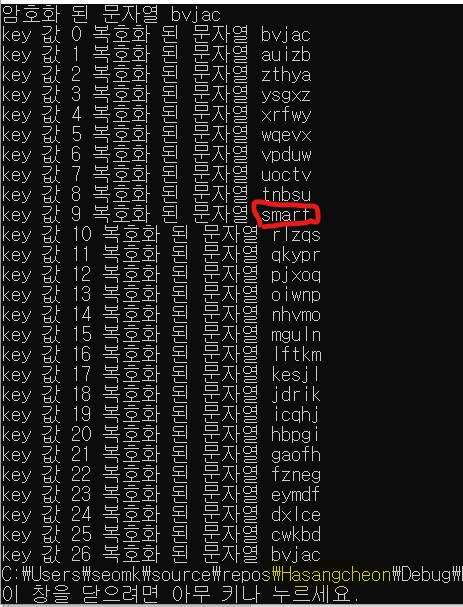
}

1. 실행화면

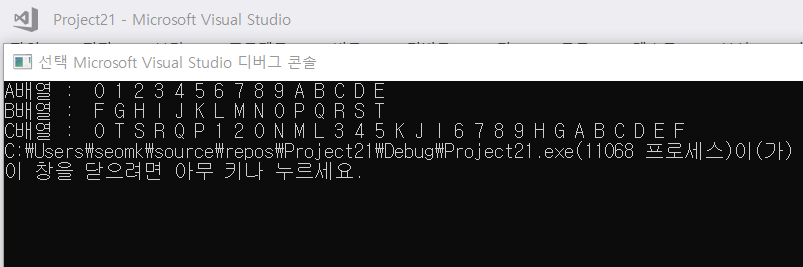


Key 값이 1부터 26사이에서 입력되어야 하는데 그렇지 않으면 다시 반복되는 사진입니다.

과제1 실행화면 입니다.



과제 1 추가문제 입니다. 답은 smart입니다.



과제 2 실행화면 입니다.

1. 고찰

과제1번은 생각보다 빨리 풀었다. 처음에는 복잡해 보이고 많아서 어려울 것 같았는데 하나하나 차근차근 아스키코드를 이용해서 푸니까 쉽게 나온 것 같다. Do...whlile문을 통해서 key값을 1부터 26까지로 제한하였고 다른 값을 입력하면 다시 입력하게 하였다. z(소문자)는 아스키코드가 122이므로 z에 key값을 더하면 26을 뺌으로써 다시 a부터 순환할 수 있게 하였다. 마찬가지로 a(소문자)의 아스키코드는 97이므로 a에 key값을 빼면 26을 더함으로써 z부터 순환할 수 있게 하였다. 과제 1의 추가문제는 암호화 된 문자 bvjac를 전수조사를 통해서 복호화 한 후 정답을 찾았다. 과제2는 코드가 바로 생각이 나지 않아서 몇시간 동안 고민을 많이 했었다. 노트에 적어가며 규칙을 찾으려고 노력하다가 생각이 났다. A배열 같은 경우는 변수 index\_a 를 0으로 초기화 한 후 변수 up을 1부터 하나씩 늘려가면서 1개,2개,3개,4개,5개를 C배열의 값으로 입력했다. index\_a도 for문을 반복할 때마다 초기값이 바뀌어야 하므로 index\_a += up;을 통해서 반복문이 끝날 때마다 index\_a에 up을 더해주었다. B배열은 A배열과 반대로 뒤에서부터 5개,4개,3개,2개,1개씩 C배열의 값으로 입력하기 때문에 A배열의 방법과 반대로 해주었다. 변수 index\_b를 14로 초기화 한 후 변수 down을 5부터 하나씩 빼 주었다. index\_b도 for문을 반복할 때마다 초기값이 바뀌어야 하므로 index\_b -= down;을 통해서 반복문이 끝날 때마다 index\_b에 down을 빼 주었다. 그리고 for문을 통해서 A, B, C배열을 출력하였다. 한칸 씩 띄어서 출력하고 싶어서 %2c를 이용하였다. for문의 초기화식에 보통 변수 = 상수 같은 식만 해보아서 처음에 변수 = 변수 라는 식을 생각하지 못한 것 같다. 이 과제를 통해서 for문에 대해서 더 많이 생각해볼 수 있었고 많은 시간을 투자해서 생각해본만큼 잊지 않고 오래 기억할 수 있을 것 같다.