

자료구조(F012) - 2장 실습 과제

G5_쓴맛볼쫄도독들(송유민, 김성준, 김현민, 김진형, 김다원)

01번 - 정답 : 4번,	02번 - 정답 : 2번,	03번 - 정답 : 4번,	04번 - 정답 : 1번,
05번 - 정답 : 3번,	06번 - 정답 : 4번,	07번 - 정답 : 1번,	08번 - 정답 : 3번,
09번 - 정답 : 2번,	10번 - 정답 : 4번,	11번 - 정답 : 4번,	12번 - 정답 : 4번,
13번 - 정답 : 3번,	14번 - 정답 : 4번,	15번 - 정답 : 2번,	

16번

<정답>

배열과 변수는 데이터를 저장하는 방식에 차이가 존재한다.

배열은 동일한 데이터 타입의 여러 값을 연속된 메모리 공간에 저장할 수 있는 자료 구조이다. 배열은 특정 크기로 선언되고, 여러 값을 저장할 수 있으며 각 값은 배열의 인덱스를 통해 접근할 수 있다.

변수는 단일 값을 저장하는 메모리 공간이다. 특정 데이터 타입에 맞는 하나의 값을 저장하고 처리할 수 있다.

즉, 변수는 하나의 값을 저장하기에 단일 메모리 공간을 사용하며, 선언된 데이터 타입에 맞는 고정크기를 갖는다. 그러나 배열은 여러 개의 값을 연속된 메모리 공간을 사용하여 저장하기에 이론상 저장 개수에 제한이 없다. 또한 인덱스를 사용하여 개별 요소에 접근이 가능하다. 그러나 변수와 동일하게 선언된 데이터 타입에 맞는 고정 크기를 갖는다.

17번

<소스코드>

```
#include <stdio.h>
#include <stdlib.h>

void count(int m[], int e[], int k[], int s[], double a[]) {
    int random;
    for (int i = 0; i < 10; i++) {
        m[i] = rand() % 101;
        e[i] = rand() % 101;
        k[i] = rand() % 101;
    }

    for (int i = 0; i < 10; i++) {
        s[i] = m[i] + e[i] + k[i];
    }

    for (int i = 0; i < 10; i++) {
        a[i] = (float)s[i] / 3;
    }

    return;
}

int main() {
    int math[10];
    int english[10];
    int korean[10];
    int sum[10];
    double ave[10];

    count(math, english, korean, sum, ave);

    printf("학번국어영어수학총점평균\n");
    printf("*****\n");
    for (int i = 0; i < 10; i++) {
        printf("%d\t%d\t%d\t%d\t%d\t%.1f\n", i + 1, korean[i], english[i], math[i], sum[i], ave[i]);
    }

    return 0;
}
```

<실행 결과>

학 번	국 어	영 어	수 학	총 점	평 균
1	72	85	41	198	66.0
2	69	80	38	187	62.3
3	96	68	65	229	76.3
4	67	49	22	138	46.0
5	63	61	51	175	58.3
6	24	66	87	177	59.0
7	71	83	80	234	78.0
8	52	64	60	176	58.7
9	49	60	90	199	66.3
10	99	23	31	153	51.0

18번

<소스코드>

```
#include <stdio.h>

int max(int n[]) {
    int tmp = 0;
    for (int i = 1; i < 5; i++) {
        if (n[tmp] < n[i]) tmp = i;
    }
    return n[tmp];
}

int min(int n[]) {
    int tmp = 0;
    for (int i = 1; i < 5; i++) {
        if (n[tmp] > n[i]) tmp = i;
    }
    return n[tmp];
}

double average(int n[]) {
    double sum = 0;
    for (int i = 0; i < 6; i++) {
        sum += n[i];
    }
    return sum / 6;
}

double dispersion(double ave, int n[]) {
    double dev[6];
    double sum = 0;
    for (int i = 0; i < 6; i++) {
        dev[i] = n[i] - ave;
    }

    for (int i = 0; i < 6; i++) {
        dev[i] += dev[i];
        sum += dev[i];
    }

    return sum / 6;
}

int main() {
    int n[6] = { 19, 2, 25, 92, 36, 45 };
    printf(" 저장된 원소 \t: ");
    for (int i = 0; i < 6; i++) {
        printf("%d ", n[i]);
    }
    printf("\n");

    printf("    최솟값 \t: %d\n", min(n));
    printf("    최댓값 \t: %d\n", max(n));
    double ave = average(n);
    printf("    평균 \t: %f\n", ave);
    printf("    분산 \t: %f\n", dispersion(ave, n));

    return 0;
}
```

<실행 결과>

```
저장된 원소      : 19  2  25  92  36  45
최솟값          : 2
최댓값          : 92
평균            : 36.500000
분산            : 796.916667
```

19번

<소스코드>

```
#include <stdio.h>

int main() {
    char a[100][8];
    int j = 0;
    for (int i = 0; i < 5; i++) {
        j = 0;
        for (a[j][i] = getchar(); a[j][i] != '\n'; a[j][i] = getchar()) {
            j++;
        }
    }

    j = 0;
    for (int i = 0; i < 5; i++) {
        while (a[j][i] != '\n') {
            printf("%c", a[j][i]);
            j++;
        }
        j = 0;
        printf("\n");
    }

    return 0;
}
```

<실행 결과>

```
asdas
safafasf
vewvv
vsdvsq
faffs
asdas
safafasf
vewvv
vsdvsq
faffs
```

20번

<소스코드>

```
#include <stdio.h>

int main() {
    char fir[100];
    char sec[100];
    int i = 0;
    for (fir[i] = getchar(); fir[i] != '\n'; fir[i] = getchar()) {
        i++;
    }
    i = 0;
    for (sec[i] = getchar(); sec[i] != '\n'; sec[i] = getchar()) {
        i++;
    }
    i = 0;
    while (1) {
        if (fir[i] != sec[i]) {
            printf("different\n");
            break;
        }
        if (fir[i] == '\n') {
            if (fir[i] == sec[i]) {
                printf("same\n");
                break;
            }
        }
        i++;
    }
    return 0;
}
```

<실행 결과>

```
as
sf
different
```

```
as
as
same
```

21번

<소스코드>

```
#include<stdio.h>
void switch_num(int* a, int* b) {
    int n1 = *a;
    *a = *b;
    *b = n1;
    return;
}
int main() {
    int num1 = 10;
    int num2 = 100;
    printf("Before switch_num : num1 == %d, num2 == %d\n", num1, num2);
    switch_num(&num1, &num2);
    printf("After switch_num : num1 == %d, num2 == %d\n", num1, num2);

    return 0;
}
```

<실행 결과>

```
Before switch_num : num1 == 10, num2 == 100
After switch_num : num1 == 100, num2 == 10
```

22번

<소스코드>

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

void print(char* arr, int p) {
    char* ptr = arr;
    for (int i = 0; i <= p - 1; i++) {
        if (*ptr >= 'A' && *ptr <= 'Z') {
            *ptr += 32;
            printf("%c", *ptr);
            ptr++;
        }
        else if (*ptr >= 'a' && *ptr <= 'z') {
            *ptr -= 32;
            printf("%c", *ptr);
            ptr++;
        }
        else {
            printf("%c", *ptr);
            ptr++;
        }
    }
}

int main() {
    char a[100];
    printf("Enter a string.\n");
    scanf("%[^\n]s", &a);
    int k = strlen(a);
    print(a, k);
    return 0;
}
```

<실행 결과>

```
Enter a string.
aAbBcCdD
AaBbCcDd
```

23번

<소스코드>

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

void print(char* arr, int t) {
    for (int i = t - 1; i >= 0; i--) {
        printf("%c", *(arr + i));
    }
}

int main() {
    char a[100];

    printf("Enter a string.\n");
    scanf("%[^\n]s", &a);

    int k = strlen(a);

    print(a, k);

    return 0;
}
```

<실행 결과>

```
Enter a string.
abcdefg
gfedcba
```

24번

<소스코드>

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

int p = 0;
typedef struct {
    char name[100];
    char phonenumber[100];
    char address[100];
    char birthday[100];
}list;

void showlist(list s[]) {
    if (p == 0) printf("목록이 없습니다.\n");
    else {
        for (int i = 0; i < p; i++) {
            printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
        }
    }
}
```

```

void add(list s[]) {

    printf("목록을 입력하세요.\n");

    printf("이름 : ");
    scanf("%s", s[p].name);

    printf("전화번호 : ");
    scanf("%s", s[p].phonenumber);

    printf("주소 : ");
    scanf("%s", s[p].address);

    printf("생일 : ");
    scanf("%s", s[p].birthday);

    printf("추가되었습니다.\n");

    p++;
}

```

```

void detect(list s[]) {

    char k[100];
    int j = 0;

    if (p == 0) {
        printf("목록이 없습니다.\n");
        return;
    }

    printf("탐색할 항목을 입력하세요.\n");
    scanf("%s", &k);

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

        if (strcmp(k, s[i].name) == 0) printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
        else if (strcmp(k, s[i].name) != 0) j++;

        if (strcmp(k, s[i].phonenumber) == 0) printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
        else if (strcmp(k, s[i].name) != 0) j++;

        if (strcmp(k, s[i].address) == 0) printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
        else if (strcmp(k, s[i].name) != 0) j++;

        if (strcmp(k, s[i].birthday) == 0) printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
        else if (strcmp(k, s[i].name) != 0) j++;
    }

    if (j == 4 + p) printf("찾으시는 항목은 없습니다.\n");
}

```

```

void delete(list s[]) {

    int k;

    if (p == 0) {
        printf("삭제할 목록이 없습니다.\n");
        return;
    }

    printf("어떤 목록을 삭제하겠습니까?\n");
    for (int i = 0; i < p; i++) {

        printf("목록 %d : %s, %s, %s, %s\n", i + 1, s[i].name, s[i].phonenumber, s[i].address, s[i].birthday);
    }

    scanf("%d", &k);
}

```



```

    if (k <= p && k > 0) {
        printf("목록 %d는 삭제되었습니다.\n", k);

        p--;
        for (int i = k; i <= p; i++) {
            s[i - 1] = s[i];
        }
    }
    else printf("목록 %d는 없습니다.\n", k);
}

int main() {
    int a;
    list sys[100];

    while (1) {
        printf("\n1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기\n입력하세요(0 입력시 프로그램 종료) : ");
        scanf("%d", &a);
        getchar();

        if (a == 0) break;
        else if (a != 0 && a != 1 && a != 2 && a != 3 && a != 4) continue;
        else if (a == 1) showList(sys);
        else if (a == 2) add(sys);
        else if (a == 3) detect(sys);
        else if (a == 4) delete(sys);
    }

    return 0;
}

```

<실행 결과>

```

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 2
목록을 입력하세요.
이름 : 송유민
전화번호 : 01022397087
주소 : 우만동
생일 : 07월08일
추가되었습니다.

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 2
목록을 입력하세요.
이름 : 김진형
전화번호 : 01022458624
주소 : 원천동
생일 : 08월27일
추가되었습니다.

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 1
목록 1 : 송유민, 01022397087, 우만동, 07월08일
목록 2 : 김진형, 01022458624, 원천동, 08월27일

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 3
탐색할 항목을 입력하세요.
송유민
목록 1 : 송유민, 01022397087, 우만동, 07월08일

```

```

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 어떤 목록을 삭제하겠습니까?
목록 1 : 송유민, 01022397087, 우만동, 07월 08일
목록 2 : 김진형, 01022458624, 원천동, 08월 27일
목록 1는 삭제되었습니다.

```

```

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 1
목록 1 : 김진형, 01022458624, 원천동, 08월 27일

```

```

1. 목록보기 || 2. 추가하기 || 3. 탐색하기 || 4. 삭제하기
입력하세요(0 입력시 프로그램 종료) : 0

```

C:\Users\A0102\source\repos\Project2\x64\Debug\Project2.exe(프로세스 19164개)이(가) 종료되었습니다

25번

<소스코드>

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int summ(int* sum, int a, int i) {
    if (i == a) {
        return *sum;
    }
    *sum += i;
    return summ(sum, a, i + 1);
}

int main() {
    int sum = 0;
    int a;
    printf("숫자를 입력해주세요\n");
    scanf("%d", &a);
    printf("%d\n", summ(&sum, a + 1, 0));

    return 0;
}

```

<실행 결과>

```

숫자를 입력해주세요
10
55

```

26번

<소스코드>

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int fibonacci(int* current, int* previous, int a, int i) {
    if (i == a) {
        return *current;
    }
    int temp = *current;
    *current += *previous;
    *previous = temp;
    return fibonacci(current, previous, a, i + 1);
}

int main() {
    int current = 1;
    int previous = 0;
    int a;
    printf("피보나치를 실행할 횟수를 입력해주세요\n");
    scanf("%d", &a);
    printf("%d\n", fibonacci(&current, &previous, a - 1, 0));
    return 0;
}

```

<실행 결과>

```
피보나치를 실행할 횟수를 입력해 주세요
7
13
```

27번

<정답>

예제 2-3에서 `getchar()`는 2개가 쓰였는데, 첫번째 `getchar`는 `scanf`가 실행되어 사용자가 값을 입력하고 엔터를 쳤을 때, 엔터(`\n`)이 버퍼에 남아있기 때문에 이것을 읽어서 지우기 위해 사용되었다. 두 번째 `getchar`는 사용자가 입력할때 까지 프로그램이 종료되지 않게 한다.