

국방정보공학과 2학년 2020032306 송민경
9주차 과제

CHAP 7)

9.

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

```
#include<math.h>
```

```
#define N 5
```

```
int main()
```

```
{
```

```
    int i, x[N] = {55, 21, 32, 47, 19};
```

```
    double average, sum = 0;
```

```
    for(i=0 ; i<N ; i++)
```

```
        sum = sum + x[i];
```

```
    average = sum/N;
```

```
    sum=0;
```

```
    for(i=0; i<N ;i++)
```

```
        sum = sum + pow(x[i]-average, 2);
```

```
    for(i=0; i<N ;i++)
```

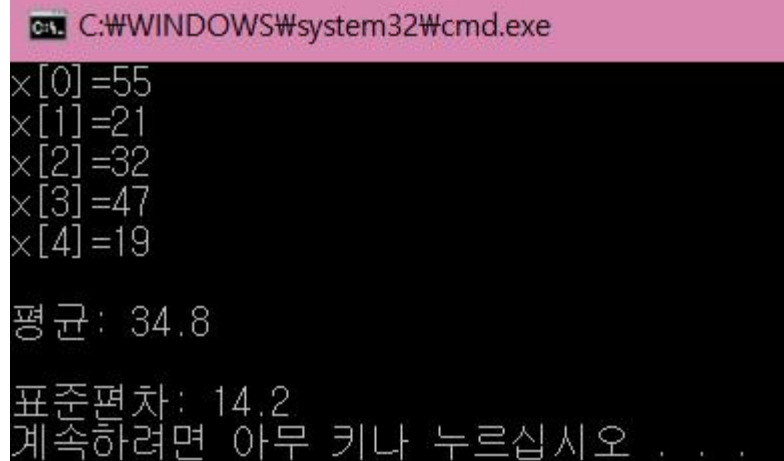
```
        printf("x[%d]=%d \n", i, x[i]);
```

```
    printf ("\n평균: %.1lf\n", average);
```

```
    printf ("\n표준편차: %.1lf\n", sqrt(sum/N));
```

```
    return 0;
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
x[0]=55
x[1]=21
x[2]=32
x[3]=47
x[4]=19

평균: 34.8
표준편차: 14.2
계속하려면 아무 키나 누르십시오 . . .
```

[소스 코드에 대한 설명]

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

```
#include<math.h>
```

```
// sqrt함수를 사용하기 위해 헤더 파일 <math.h>를 포함시킨다.
```

```
#define N 5
```

```
int main()
```

```
{
```

```
    int i, x[N] = {55, 21, 32, 47, 19};
```

```
    double average, sum = 0;
```

```
    // for문을 활용해 평균을 구한다.
```

```
    for(i=0 ; i<N ; i++)
```

```
        sum = sum + x[i];
```

```
    average = sum/N;
```

```
    // for문을 활용해 표준편차를 구한다.
```

```
    sum=0;
```

```
    for(i=0; i<N ;i++)
```

```
        sum = sum + pow(x[i]-average, 2);
```

```
    // for문을 활용해 배열원소를 출력한다.
```

```
    for(i=0; i<N ;i++)
```

```
        printf("x[%d]=%d \n", i, x[i]);
```

```
    // 평균과 표준편차를 출력한다.
```

```
    printf ("\n평균: %.1lf\n", average);
```

```
    printf ("\n표준편차: %.1lf\n", sqrt(sum/N));
```

```
    return 0;
```

```
}
```

10.

```
#include <stdio.h>
#include <string.h>
#define N 10

int main()
{
    char name[N][9] = {"홍길동", "이몽룡", "성춘향", "제갈공명", "주몽",
                      "배장화", "배홍련", "홍길동", "연홍부", "연놀부"};
    char phone[N][14] = {"010-5210-1234", "010-523-1628",
                        "010-1235-8765", "010-345-1676", "010-5210-5463",
                        "010-523-7979", "010-5210-1472", "010-8255-8255",
                        "010-516-3483", "010-8282-8282"};
    double grade[N] = {4.2, 4.0, 3.2, 4.4, 3.7, 2.9, 4.0, 3.8, 3.2, 2.7};
    int order[N];
    int index, temp, i, repeat;
    char swap;

    for (i=0; i<N; i++)
        order[i] = i;

    for (repeat=1; repeat<N; repeat++)
    {
        swap = 'N';
        for (i=0; i<N-repeat; i++)
        {
            if (strcmp(name[order[i]], name[order[i+1]]) > 0)
            {
                temp = order[i];
                order[i] = order[i+1];
                order[i+1] = temp;

                swap = 'Y';
            }
        }
        if (swap == 'N')
            break;
    }

    printf(" 이름      전화번호      평점 \n");
    printf("===== \n");
    for (i=0; i<N; i++)
    {
```

```

        index = order[i];
        printf("%-10s  %-14s  %5.1lf  \n",      name[index],  phone[index],
grade[index]);

        if ((i+1)%5 ==0)
            printf("-----\n");
    }

```

| 이름 | 전화번호 | 평점 |
|------|---------------|-----|
| 배장화 | 010-523-7979 | 2.9 |
| 배홍련 | 010-5210-1472 | 4.0 |
| 성춘향 | 010-1235-8765 | 3.2 |
| 연놀부 | 010-8282-8282 | 2.7 |
| 연흥부 | 010-516-3483 | 3.2 |
| 이몽룡 | 010-523-1628 | 4.0 |
| 제갈공명 | 010-345-1676 | 4.4 |
| 주몽 | 010-5210-5463 | 3.7 |
| 홍길동 | 010-5210-1234 | 4.2 |
| 홍길 | 010-8255-8255 | 3.8 |

계속하려면 아무 키나 누르십시오 . . .

[소스 코드에 대한 설명]

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

```
#include <string.h>
```

```
#define N 10
```

```
int main()
```

```
{
```

```
    char name[N][9] = {"홍길동", "이몽룡", "성춘향", "제갈공명", "주몽",
                        "배장화", "배홍련", "홍길동", "연흥부", "연놀부"};
```

```
    char phone[N][14] = {"010-5210-1234", "010-523-1628",
                        "010-1235-8765", "010-345-1676", "010-5210-5463",
                        "010-523-7979", "010-5210-1472", "010-8255-8255",
                        "010-516-3483", "010-8282-8282"};
```

```
    double grade[N] = {4.2, 4.0, 3.2, 4.4, 3.7, 2.9, 4.0, 3.8, 3.2, 2.7};
```

```
    int order[N];
```

```
    int index, temp, i, repeat;
```

```
    char swap; //이웃한 두 원소의 값이 교환되었는지를 알려주는 플래그 변수
```

```

    for (i=0; i<N; i++)
        order[i] = i;

//버블 정렬을 이용해 정렬한다.
for (repeat=1; repeat<N; repeat++)
{
    swap = 'N';
    for (i=0; i<N-repeat; i++)
    {
        if (strcmp(name[order[i]], name[order[i+1]]) > 0)
        {
            temp = order[i];
            order[i] = order[i+1];
            order[i+1] = temp;

            swap = 'Y';
        }
    }
    // 두 원소 간 교환이 발생하지 않았다면, 정렬작업을 끝낸다.
    if (swap == 'N')
        break;
}

//정렬 결과를 출력한다.
printf(" 이름      전화번호      평점 \n");
printf("===== \n");
for (i=0; i<N; i++)
{
    index = order[i];
    printf("%-10s  %-14s  %5.1lf  \n",      name[index],  phone[index],
grade[index]);

    if ((i+1)%5 ==0)
        printf("----- \n");
}

return 0;
}

```

11.

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

```
#define L 2
```

```
#define M 3
```

```
#define N 4
```

```
int main()
```

```
{
```

```
    int A[L][M] = {{1, 2, 3}, {4, 5, 6}};
```

```
    int B[M][N] = {{3, 4, 5, 6}, {1, 2, 4, 3}, {4, 2, 6, 1}};
```

```
    int C[L][N];
```

```
    int i, j, k;
```

```
    for(i=0; i<L; i++)
```

```
        for(j=0; j<N; j++)
```

```
        {
```

```
            C[i][j]=0;
```

```
            for (k=0; k<M; k++)
```

```
                C[i][j] += A[i][k] * B[k][j];
```

```
        }
```

```
    printf("\n 행렬 A\n");
```

```
    for(i=0; i<L; i++)
```

```
    {
```

```
        printf(" [ ");
```

```
        for(j=0; j<M; j++)
```

```
            printf("%2d ", A[i][j]);
```

```
        printf("] \n");
```

```
    }
```

```
    printf("\n 행렬 B\n");
```

```
    for(i=0; i<M; i++)
```

```
    {
```

```
        printf(" [ ");
```

```
        for(j=0; j<N; j++)
```

```
            printf("%2d ", B[i][j]);
```

```
        printf("] \n");
```

```
    }
```

```
    printf("\n 행렬 C=A*B \n");
```

```
    for(i=0; i<L; i++)
```

```
    {
```

```

        printf(" [ ");
        for(j=0; j<N; j++)
            printf("%2d ", C[i][j]);
        printf("] \n");
    }

    return 0;
}

```

The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The output is as follows:

```

행렬 A
[ 1 2 3 ]
[ 4 5 6 ]

행렬 B
[ 3 4 5 6 ]
[ 1 2 4 3 ]
[ 4 2 6 1 ]

행렬 C=A*B
[ 17 14 31 15 ]
[ 41 38 76 45 ]
계속하려면 아무 키나 누르십시오 . . .

```

[소스 코드에 대한 설명]

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

```
#define L 2
```

```
#define M 3
```

```
#define N 4
```

```
int main()
```

```
{
```

```
    int A[L][M] = {{1, 2, 3}, {4, 5, 6}};
```

```
    int B[M][N] = {{3, 4, 5, 6}, {1, 2, 4, 3}, {4, 2, 6, 1}};
```

```
    int C[L][N];
```

```
    int i, j, k;
```

```
// 배열 C에 두 행렬의 곱을 구한다.
```

```
    for(i=0; i<L; i++)
```

```
        for(j=0; j<N; j++)
```

```
        {
```

```
            C[i][j]=0;
```

```

        for (k=0; k<M; k++)
            C[i][j] += A[i][k] * B[k][j];
    }
//행렬 A를 출력한다.
printf("\n 행렬 A\n");
for(i=0; i<L; i++)
{
    printf(" [ ");
    for(j=0; j<M; j++)
        printf("%2d ", A[i][j]);
    printf("] \n");
}

//행렬 B를 출력한다.
printf("\n 행렬 B\n");
for(i=0; i<M; i++)
{
    printf(" [ ");
    for(j=0; j<N; j++)
        printf("%2d ", B[i][j]);
    printf("] \n");
}

//행렬 C를 출력한다.
printf("\n 행렬 C=A*B \n");
for(i=0; i<L; i++)
{
    printf(" [ ");
    for(j=0; j<N; j++)
        printf("%2d ", C[i][j]);
    printf("] \n");
}

return 0;
}

```


12.

```
#include <stdio.h>
#define STUDENT 10
#define N 10
#define YES 1
#define NO 0

int main()
{
    int paper[STUDENT][N]= {{1, 3, 2, 3, 4, 2, 3, 1, 4, 3},
                             {1, 2, 2, 2, 4, 2, 3, 1, 4, 2},
                             {4, 3, 2, 3, 4, 2, 3, 1, 4, 2},
                             {1, 3, 2, 2, 4, 3, 3, 4, 4, 2},
                             {1, 3, 2, 3, 4, 2, 3, 3, 4, 1},
                             {1, 1, 2, 4, 4, 3, 3, 2, 4, 3},
                             {1, 3, 2, 3, 4, 2, 3, 1, 4, 2},
                             {1, 3, 2, 3, 3, 2, 3, 1, 4, 2},
                             {2, 3, 3, 3, 3, 2, 4, 1, 4, 2},
                             {3, 4, 4, 2, 4, 1, 2, 1, 4, 2} };
    int answer[N]={1, 3, 2, 3, 4, 2, 3, 1, 4, 2};
    int total[STUDENT]= {0}, id[STUDENT], rank[STUDENT];
    int i, j, u, std, pass, temp, swap;

    for (i=0; i<STUDENT; i++)
    {
        id[i]=i+1;
    }

    printf("\n\t ■ 문항별 채점 결과 ■ \n");
    printf("===== \n");
    printf(" 문항");
    for (i=0; i<N; i++)
        printf("%4d", i);
    printf(" 점수");
    printf("\n===== \n");

    for (std=0; std<STUDENT; std++)
    {
        printf("%3d번 ", std+1);

        for (i=0; i<N; i++)
            if (paper[std][i] == answer[i])
            {
```

```

        total[std]++;
        printf("   ○");
    }
    else
        printf("   ×");

    printf(" %3d점 \n", total[std]);
}
printf("=====\\n");

for (pass=1; pass<STUDENT; pass++)
{
    swap=NO;
    for (i=0; i<STUDENT-pass; i++)
        if (total[i] < total[i+1])
        {

            temp = total[i];
            total[i] = total[i+1];
            total[i+1] = temp;

            temp = id[i];
            id[i] = id[i+1];
            id[i+1] = temp;

            swap=YES;
        }

    if (swap==NO) break;
}

for (i=0; i<STUDENT; i++)
{
    rank[i] = 1;
    for (u=0; u<STUDENT; u++)
        if (total[i] < total[u])
            rank[i]++;
}

printf("\\n\\t ■ 정렬 및 석차 ■ \\n");

```

```

printf("\t =====\n");
printf("\t  번호  점수  석차 \n");
printf("\t =====\n");
for ( i = 0 ; i< STUDENT; i++ )
    printf("\t  %2d번  %2d점  %2d등 \n", id[i], total[i], rank[i]);
printf("\t =====\n\n");

return 0;
}

```

The screenshot shows a Windows command prompt window with the title "C:\WINDOWS\system32\cmd.exe". The output displays a table of quiz results for 10 students, followed by a section titled "정렬 및 석차" (Sorting and Ranking) showing the same data sorted by score in descending order.

| 번호 | 점수 | 석차 |
|-----|-----|-----|
| 1번 | 9점 | 1등 |
| 2번 | 8점 | 2등 |
| 3번 | 9점 | 2등 |
| 4번 | 7점 | 7등 |
| 5번 | 8점 | 5등 |
| 6번 | 5점 | 9등 |
| 7번 | 10점 | 1등 |
| 8번 | 9점 | 2등 |
| 9번 | 6점 | 8등 |
| 10번 | 4점 | 10등 |

계속하려면 아무 키나 누르십시오 . . .

[소스 코드에 대한 설명]

```

#include <stdio.h>
#define STUDENT 10
#define N 10
#define YES 1
#define NO 0

int main()
{
    int paper[STUDENT][N]= {{1, 3, 2, 3, 4, 2, 3, 1, 4, 3},
                             {1, 2, 2, 2, 4, 2, 3, 1, 4, 2},
                             {4, 3, 2, 3, 4, 2, 3, 1, 4, 2},
                             {1, 3, 2, 2, 4, 3, 3, 4, 4, 2},
                             {1, 3, 2, 3, 4, 2, 3, 3, 4, 1},
                             {1, 1, 2, 4, 4, 3, 3, 2, 4, 3},
                             {1, 3, 2, 3, 4, 2, 3, 1, 4, 2},
                             {1, 3, 2, 3, 3, 2, 3, 1, 4, 2},
                             {2, 3, 3, 3, 3, 2, 4, 1, 4, 2},
                             {3, 4, 4, 2, 4, 1, 2, 1, 4, 2} };

    int answer[N]={1, 3, 2, 3, 4, 2, 3, 1, 4, 2};
    int total[STUDENT]= {0}, id[STUDENT], rank[STUDENT];
    int i, j, u, std, pass, temp, swap;

    // id배열에 학생의 번호를 저장한다.
    for (i=0; i<STUDENT; i++)
    {
        id[i]=i+1;
    }

    // 문항별 채점 결과를 출력한다.
    printf("\n\t ■ 문항별 채점 결과 ■ \n");
    printf("===== \n");
    printf(" 문항");
    for (i=0; i<N; i++)
        printf("%4d", i);
    printf(" 점수");
    printf("\n===== \n");

    // 학생별 답안과 정답을 비교하여 채점 결과를 출력한다.
    for (std=0; std<STUDENT; std++)
    {
        printf("%3d번 ", std+1);
    }

```

```

        for (i=0; i<N; i++)
            if (paper[std][i] == answer[i])
            {
                total[std]++;
                printf("  ○");
            }
            else
                printf("  ×");

        printf(" %3d점 \n", total[std]);
    }
    printf("===== \n");

// 버블 정렬을 이용해 점수를 정렬한다.
for (pass=1; pass<STUDENT; pass++)
{
    swap=NO;
    for (i=0; i<STUDENT-pass; i++)
        if (total[i] < total[i+1])
        {

            temp = total[i];
            total[i] = total[i+1];
            total[i+1] = temp;

            temp = id[i];
            id[i] = id[i+1];
            id[i+1] = temp;

            swap=YES;
        }

    if (swap==NO) break;
    // 두 원소 간 교환이 발생하지 않았다면, 정렬작업을 끝낸다.
}

// 석차를 구한다.
// 나의 석차를 1로 초기화한 후, 나보다 점수가 높은 학생 수가 있으면 1 증가한다.
for (i=0; i<STUDENT; i++)
{
    rank[i] = 1;

```

```

        for (u=0; u<STUDENT; u++)
            if (total[i] < total[u])
                rank[i]++;

    }

    // 결과를 출력한다.
    printf("\n\t ■ 정렬 및 석차 ■ \n");
    printf("\t =====\n");
    printf("\t 번호  점수  석차 \n");
    printf("\t =====\n");
    for ( i = 0 ; i< STUDENT; i++ )
        printf("\t  %2d번  %2d점 %2d등 \n", id[i], total[i], rank[i]);
    printf("\t =====\n\n");

    return 0;
}

```