



R E P O R T

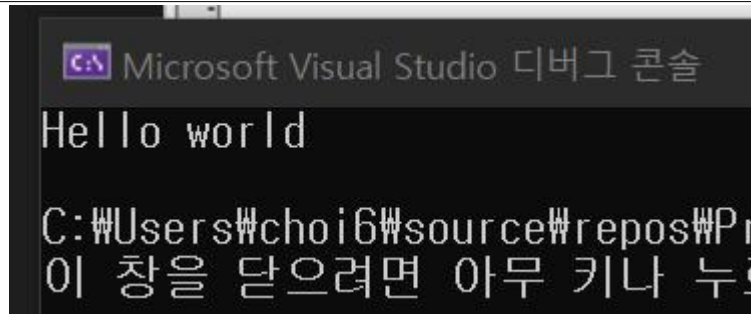
C프로그래밍2 과제2

과목명	C 프로그래밍 II
분반	2 분반
교수	정 구 철
학번	2020136129
이름	최 수 연
제출일	2020년 9월 16일 수요일

1. char sentence1에 “Hello world” 입력하고 출력하시오.

```
#include <stdio.h>

int main(void)
{
    char sentence1[11] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd' };
    for (int i = 0; i < 11; i++)
    {
        printf("%c", sentence1[i]);
    }
    printf("\n");
    return 0;
}
```



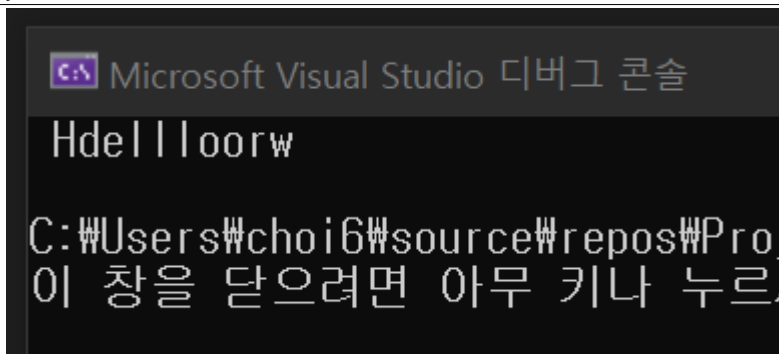
2. 'Hello world'를 아스키코드에 의거해 오름차순으로 정렬하고 출력하시오.

□ Hint, 결과는 'Hde||loorw'와 같이 출력되어야함

```
#include <stdio.h>

int main(void)
{
    char temp, sentence1[11] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd' };
    int i, j, least;

    for (i = 0; i < 10; i++)
    {
        least = i;
        for (int j = i + 1; j < 11; j++)
            if (sentence1[j] < sentence1[least])
                least = j;
        temp = sentence1[i];
        sentence1[i] = sentence1[least];
        sentence1[least] = temp;
    }
    for (int i = 0; i < 11; i++)
        printf("%c", sentence1[i]);
    printf("\n");
    return 0;
}
```



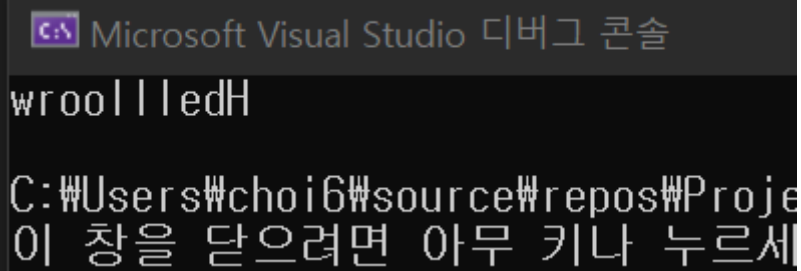
3. 'Hello world'를 아스키코드에 의거해 내림차순으로 정렬하고 출력하시오

☐ Hint, 결과는 'wroolllledH'와 같이 출력되어야함

```
#include <stdio.h>

int main(void)
{
    char temp, sentence1[11] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd' };
    int i, j, max;

    for (i = 0; i < 10; i++)
    {
        max = i;
        for (int j = i + 1; j < 11; j++)
            if (sentence1[max] < sentence1[j])
                max = j;
        temp = sentence1[i];
        sentence1[i] = sentence1[max];
        sentence1[max] = temp;
    }
    for (int i = 0; i < 11; i++)
        printf("%c", sentence1[i]);
    printf("\n");
    return 0;
}
```



Microsoft Visual Studio 디버그 콘솔

wroolllledH

C:\Users\choi6\source\repos\Proje
이 창을 닫으려면 아무 키나 누르세

4. 특정 문자가 있는지 확인하는 함수를 만드시오. 그리고 이 함수를 이용하여 “Hello world”에 ‘o’가 있는지 없는지에 대한 결과를 출력하시오.

- ☐ 함수의 원형은 int hasChar(char sentence[], int len, char find);으로 선언
- ☐ 반환 값은 참이면 1, 거짓이면 0을 반환
- ☐ 출력은 ‘o’가 존재하면 ‘o is exist’, 존재하지 않으면 ‘o is not exist’로 출력

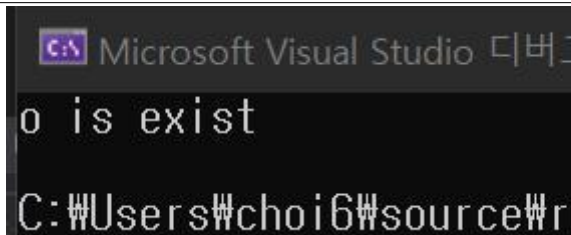
```
#include <stdio.h>

int hasChar(char sentence[], int len, char find);

int main(void)
{
    char sentence1[11] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd' };
    char find = 'o';
    int len = 0, x = 0;

    x = hasChar(sentence1, 11, find);
    if (x == 1)
        printf("o is exist\n");
    else
        printf("o is not exist\n");
}

int hasChar(char sentence[], int len, char find)
{
    int x = 0;
    for (int i = 0; i < len; i++)
    {
        if (sentence[i] == find)
            x++;
    }
    if (x != 0)
        return 1;
    else
        return 0;
}
```



5. 특정 문자가 있는지 확인하는 함수를 만드시오. 단 이 함수는 **이진탐색 알고리즘**을 사용해야 함. 그리고 이 함수를 이용하여 “Hello world”에 ‘o’가 있는지 없는지에 대한 결과를 출력하시오.

- ☐ 함수의 원형은 int hasChar_binary(char sentence[], int len, char find);으로 선언
- ☐ 반환 값은 참이면 1, 거짓이면 0을 반환
- ☐ 출력은 ‘o’가 존재하면 ‘o is exist’, 존재하지 않으면 ‘o is not exist’로 출력
- ☐ hint, 해당 함수는 오름차순 정렬 알고리즘도 내포하고 있어야 함

```
#include <stdio.h>

int hasChar_binary(char sentence[], int len, char find);

int main(void)
{
    char sentence1[11] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd' };
    char find = 'o';
    int len = 0, x = 0;

    int i, least, temp;

    for (i = 0; i < 10; i++)
    {
        least = i;
        for (int j = i + 1; j < 11; j++)
            if (sentence1[j] < sentence1[least])
                least = j;
        temp = sentence1[i];
        sentence1[i] = sentence1[least];
        sentence1[least] = temp;
    }

    x = hasChar_binary(sentence1, 11, find);
    if (x == 1)
        printf("o is exist\n");
    else
        printf("o is not exist\n");
}

int hasChar_binary(char sentence[], int len, char find)
{
    int x = 0, low = 0, high = len - 1, middle;
```

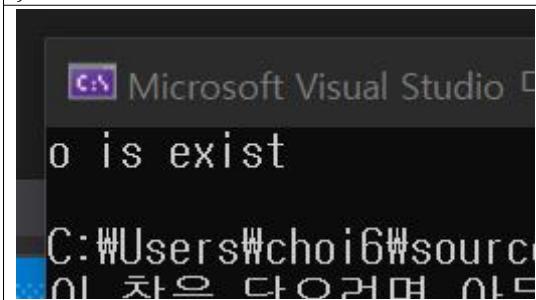
```

while (low <= high)
{
    middle = (low + high) / 2;

    if (sentence[middle] < find)
        low = middle + 1;
    else if (sentence[middle] > find)
        high = middle - 1;
    else if (sentence[middle] == find)
        return 1;
}

if (x == 1)
    return 1;
else
    return 0;
}

```



6. $A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$ 를 정수형 이차원 배열로 정의하고 출력하시오

```
#include <stdio.h>

int main(void)
{
    int A[2][2] = { {1, 2}, {2, 3} };

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
            printf("%d ", A[i][j]);
        printf("\n");
    }
    return 0;
}
```

A screenshot of a terminal window showing the output of the C program. The output is a 2x2 matrix of integers: 1 2 on the first line and 2 3 on the second line. The terminal has a dark background with light-colored text. The Microsoft logo is visible in the top left corner of the terminal window.

```
1 2
2 3
```


7. 정수형 이차원 배열을 출력하는 함수를 만들고 $A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ 를 출력하시오.

☐ #define SIZE 2

☐ 함수의 원형은 void printMatrix(int mat[SIZE][SIZE])로 정의

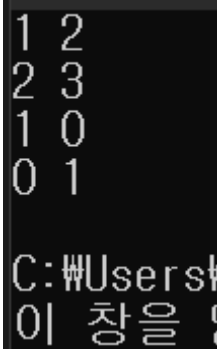
```
#include <stdio.h>
#define SIZE 2

void printMatrix(int mat[SIZE][SIZE]);

int main(void)
{
    int A[SIZE][SIZE] = { {1, 2}, {2, 3} };
    int B[SIZE][SIZE] = { {1, 0}, {0, 1} };
    printMatrix(A);
    printMatrix(B);

    return 0;
}

void printMatrix(int mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}
```



```
1 2
2 3
1 0
0 1

C:\Users\st
이 창을 열
```

8. 정수형 행렬을 더하는 함수를 만들고 $A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ 의 합을 구하시오. 합한 값은 정수형 행렬 C에 저장하고 C를 출력하시오. 출력 시에 문제 7에서 만든 printMatrix를 사용하시오

☐ #define SIZE 2

☐ 함수의 원형은 void sumMatrix(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int sum_mat[SIZE][SIZE])로 정의

```
#include <stdio.h>
#define SIZE 2

void printMatrix(int mat[SIZE][SIZE]);
void sumMatrix(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int sum_mat[SIZE][SIZE]);

int main(void)
{
    int A[SIZE][SIZE] = { {1, 2}, {2, 3} };
    int B[SIZE][SIZE] = { {1, 0}, {0, 1} };
    int C[SIZE][SIZE];

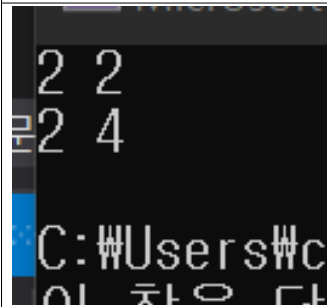
    sumMatrix(A, B, C);
    printMatrix(C);

    return 0;
}

void printMatrix(int mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}

void sumMatrix(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int sum_mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
```

```
        for (int j = 0; j < SIZE; j++)  
        {  
            sum_mat[i][j] = mat1[i][j] + mat2[i][j];  
        }  
    }  
}
```



```
2 2  
2 4  
C:\Users\c  
이 차은 다
```

9. 2차원 배열을 1차원 배열로 만드는 함수와 1차원 배열을 2차원 배열로 만드는 함수를 만드시오. 변환할 배열은 $A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$ 이고 A를 2차원에서 1차원으로 변환 후 값을 출력하고 1차원에서 2차원 변환 후 값을 출력하시오

- ☐ #define SIZE 2로 SIZE 정의
- ☐ 함수의 원형은 void mat2vec(int mat[SIZE][SIZE], int vec[]), void vec2mat(int mat[SIZE][SIZE], int vec[])
- ☐ 2차원 배열 출력시 문제 7에서 만든 printMatrix를 사용하시오
- ☐ 1차원 배열 출력은 자유롭게 해도 됨

```
#include <stdio.h>
#define SIZE 2

void printMatrix(int mat[SIZE][SIZE]);
void mat2vec(int mat[SIZE][SIZE], int vec[]);
void vec2mat(int mat[SIZE][SIZE], int vec[]);

int main(void)
{
    int A[SIZE][SIZE] = { {1, 2}, {2, 3} };
    int B[SIZE * SIZE];
    int C[SIZE][SIZE];

    mat2vec(A, B);

    for (int i = 0; i < SIZE * SIZE; i++)
    {
        printf("%d ", B[i]);
    }
    printf("\n\n");
    vec2mat(C, B);
    printMatrix(C);

    return 0;
}

void printMatrix(int mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
            printf("%d ", mat[i][j]);
```

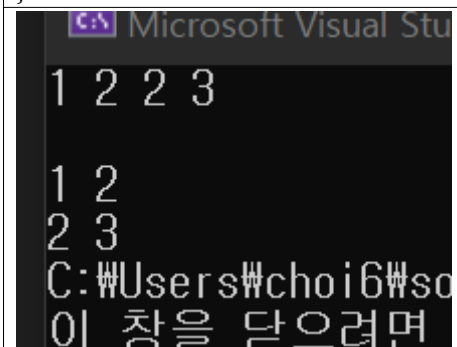
```

        printf("\n");
    }
}

void mat2vec(int mat[SIZE][SIZE], int vec[])
{
    int o = 0;
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++, o++)
        {
            vec[o] = mat[i][j];
        }
    }
}

void vec2mat(int mat[SIZE][SIZE], int vec[])
{
    int o = 0;
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++, o++)
        {
            mat[i][j] = vec[o];
        }
    }
}

```



10. $A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 3 & 2 \\ 1 & 3 \end{pmatrix}$ 를 더하고 그 결과를 오름차순 정렬하시오. 정렬된 값은 정수형 행렬 C에 저장하고 C를 출력하시오.

- ☐ #define SIZE 2로 SIZE 정의
- ☐ 함수 원형은 void sumAlign(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int align_mat[SIZE][SIZE])로 정의
- ☐ sumAlign에서 행렬간 합은 문제 8에서 만든 sumMatrix를 사용하시오.
- ☐ 출력 시 문제 7에서 만든 printMatrix를 사용하시오
- ☐ hint, 최종 결과물은 $C = \begin{pmatrix} 3 & 4 \\ 4 & 6 \end{pmatrix}$ 와 같이 나와야함
- ☐ hint2, 9에서 사용한 함수들을 응용해도 됨

```
#include <stdio.h>
#define SIZE 2

void printMatrix(int mat[SIZE][SIZE]);
void mat2vec(int mat[SIZE][SIZE], int vec[]);
void vec2mat(int mat[SIZE][SIZE], int vec[]);
void sumMatrix(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int sum_mat[SIZE][SIZE]);
void sumAlign(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int align_mat[SIZE][SIZE]);

int main(void)
{
    int A[SIZE][SIZE] = { {1, 2}, {2, 3} };
    int B[SIZE][SIZE] = { {3, 2}, {1, 3} };
    int C[SIZE][SIZE];

    sumAlign(A, B, C);

    return 0;
}

void printMatrix(int mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}
```

```

void mat2vec(int mat[SIZE][SIZE], int vec[])
{
    int o = 0;
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++, o++)
        {
            vec[o] = mat[i][j];
        }
    }
}

void vec2mat(int mat[SIZE][SIZE], int vec[])
{
    int o = 0;
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++, o++)
        {
            mat[i][j] = vec[o];
        }
    }
}

void sumMatrix(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int
sum_mat[SIZE][SIZE])
{
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
        {
            sum_mat[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}

void sumAlign(int mat1[SIZE][SIZE], int mat2[SIZE][SIZE], int
align_mat[SIZE][SIZE])
{
    int a[SIZE][SIZE], b[SIZE * SIZE], c;
    sumMatrix(mat1, mat2, a);
    mat2vec(a, b);
}

```

```
for (int i = 0; i < SIZE * SIZE - 1; i++)
{
    for (int j = i + 1; j < SIZE * SIZE - 1; j++)
    {
        if (b[i] > b[j])
        {
            c = b[i];
            b[i] = b[j];
            b[j] = c;
        }
    }
}
vec2mat(aligned_mat, b);
printMatrix(aligned_mat);
}
```

