

Programming Practice 09

Pointer, Array, String

Contents

- Example (50 min)
- Questions (60 min)
- Final Term (40min)

Contents

- **Example (50 min)**
- Questions (60 min)
- Final Term (40min)

Example (1)

```
#include <stdio.h>

void main( void ){
    char *cP;
    short *sP;
    int *iP;

    printf( "char포인터 크기 %d, short포인터 크기 %d, int포인터 크기 %dn Wn",
           sizeof( cP ), sizeof( sP ), sizeof( iP ) );
}
```

Example (2)

```
#include <stdio.h>

int main(void){

    int a = 10;
    int b = 20;
    int* p1 = &a;
    int* p2 = &b;
    //--연산자가 *연산자보다 우선순위가기 때문에 반드시 괄호로 묶어야 함
    --(*p1);
    --(*p2);
    //가리키는 대상을 바꿔줌
    p1 = &b;
    p2 = &a;
    //*p1과 *p2에서 *연산자를 빼고 출력하면, 메모리의 주소값이 출력됨
    //*연산자를 붙여줘야 가리키고있는 실제 값이 출력
    printf("%d\n", *p1);
    printf("%d\n", *p2);
    return 0;
}
```

Example (3)

```
#include <stdio.h>

//배열을 인자로 받아 홀수를 출력해주는 프로그램
void oNum(int* ptr){

    printf("홀수 출력: ");

    //홀수를 찾기 위해 반복문 수행
    for (int i = 0; i < 10; i++)
        if (ptr[i] % 2 != 0) //배열값을 나눴을때 나머지가 0이 아니면 홀수
            printf("%d ", ptr[i]);
}

void eNum(int* ptr){

    printf("짝수 출력: ");

    //짝수를 찾기 위해 반복문 수행
    for (int i = 0; i < 10; i++)
        if (ptr[i] % 2 == 0) //배열값을 나눴을때 나머지가 0이면 짝수
            printf("%d ", ptr[i]);
}
```

```
int main(void){

    int arr[10];
    int input;

    printf("총 10개의 숫자 입력 \n");

    //10개의 숫자를 입력받기 위해 반복문을 수행
    for (int i = 0; i < 10; i++){
        printf("?: ");
        scanf_s("%d", &input);
        arr[i] = input;
    }

    oNum(arr); //배열을 oNum에 인자로 전달
    printf("\n");
    eNum(arr); //배열을 eNum에 인자로 전달
    printf("\n");

    return 0;
}
```

Example (4)

```
#include <stdio.h>

int main(void){

    int input;
    int arr[1000]; //배열크기는 적당히 정해도 상관 없음
    int count = 0;
    printf("10진수 정수 입력 : ");
    scanf_s("%d", &input);

    /*배열에 input을 2로나눈 나머지값을 차례대로 저장(2진수 값이 거꾸로 저장됨)
    반복문의 조건을 input>1로 하게되면 제일 마지막에 값을 나눈 몫을 저장하지 못하므로
    0이 될 때 까지반복함*/
    for (int i = 0; input > 0; i++){
        arr[i] = input % 2;
        input = input / 2;
        count++; //count를 증가하여 후에 배열을 몇 번 반복할지를 결정
    }

    //배열에 저장된 값을 거꾸로 출력하여 정상적인 2진수 값 출력
    for (int i = count-1; i >= 0; i--)
        printf("%d ", arr[i]);

    return 0;
}
```

Example (5)

```
#include <stdio.h>

int main(void)
{
    char str01[] = "This is a string.";    // 크기를 지정하지 않은 문자열 변수 선언
    char str02[7] = "string";              // 크기를 지정한 문자열 변수 선언

    printf("이것은 문자열 상수입니다.\n"); // 문자열 상수
    printf("문자열 str01에 저장되어 있는 문자열은 W"%sW"입니다.\n", str01);
    printf("문자열 str02에 저장되어 있는 문자열은 W"%sW"입니다.\n", str02);
    return 0;
}
```


Example (6)

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str[20];
    char ch;
    while (1)
    {
        puts("미국의 수도를 입력하세요 :");
        scanf("%s", str);
        // strcmp 함수 == 문자열의 비교
        if (strcmp(str, "워싱턴") == 0 || strcmp(str, "washington") == 0){
            puts("정답입니다!");
            break;
        }
        else
            puts("아쉽네요~");
        fflush(stdin); //버퍼 초기화
```

```
        puts("\n이 프로그램을 끝내고자 한다면 'q'를 눌러주세요!");
        puts("계속 도전하고자 하시면 Enter를 눌러주세요!");
        scanf("%c", &ch);
        if (ch == 'q'){// 조건문을 이용한 단일문자의 비교
            break;
        }
        fflush(stdin);
    }
    return 0;
}
```

Contents

- Example (50 min)
- **Questions (60 min)**
- Final Term (40min)

Questions(1)

- 입력값이 홀수면 앞부터, 짝수면 뒤부터 저장하는 프로그램
- 조건
 1. 길이가 10인 배열을 선언하고 10개의 정수를 입력 받음
 1. 입력받은 숫자가 홀수면 배열의 앞부터, 짝수면 뒤부터 채워서 출력
- Key Point
 1. 10개의 숫자를 입력받기 위해 반복문을 수행
 2. 배열의 순서를 제어하기 위한 Index로 사용할 변수가 필요함

Questions(1)

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

```
int main(void){
```

```
    int arr[10];
```

```
    int input;
```

```
    int front = 0; //앞 순서 인덱스
```

```
    int rear = 9; //뒤 순서 인덱스
```

```
    printf("총 10개의 숫자 입력 \n");
```

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

```
        printf("?: ");
```

```
        scanf_s("%d", &input);
```

```
        //사용자가 입력한 값이 홀수면 앞에서부터 채워나감
```

```
        if (input % 2 != 0)
```

```
            //가장 앞에 저장 후 front값을 1증가하여 다음번에는 그 뒤에 저장
```

```
            arr[front++] = input;
```

```
        //사용자가 입력한 값이 짝수면 뒤에서부터 채워나감
```

```
        else
```

```
            //가장 뒤에 저장 후 rear값을 1감소하여 다음번에는 그 앞에 저장
```

```
            arr[rear--] = input;
```

```
    }
```

```
    printf("배열 요소의 출력: ");
```

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

```
        printf("%d ", arr[i]);
```

```
    return 0;
```

```
}
```

Questions(2)

- 문자가 나타나는 빈도를 계산하여 출력하는 프로그램
- 사용자로부터 문자열을 입력 받은 후(알파벳 소문자 20글자까지) 그 빈도를 출력한다.
- 문자열에서 A~Z까지 총 몇 번 입력되었는지 보여준다.
- Key Point
 1. 문자열의 각 문자에 접근하기 위한 방법이 필요
 2. 문자열을 검사하고 해당 문자의 개수를 1씩 증가
 3. `for(i='a'; i<='z'; i++)` 를 사용하여 `%c`로 `i`를 출력하면 A~Z를 쉽게 출력 가능

Questions(2)

```
#include <stdio.h>
int str_chr(char* s, int c);
void count_ctr(char* s);

int main() {
    char str[20];
    char ctr;

    printf("문자열을 입력하시오: ");
    gets(str);
    count_ctr(str);

    return 0;
}
```

```
int str_chr(char* s, char c) {
    int i, count=0;
    for(i=0; s[i] != NULL; i++) {
        if(s[i] == c)
            count++;
    }
    return count;
}

void count_ctr(char* s) {
    int i;
    for(i='a'; i<='z'; i++)
        printf("%c : %d\n", i, str_chr(s, i));
}
```

Questions(3)

- 회문검사 프로그램
- 회문이란?
 - abba, noon, akka, eee, ere 등등 앞 뒤를 바꿔도 동일한 문자가 되는 것을 말함
- Key Point
 1. 우선 회문을 검사하려면, 사용자가 입력한 문자열의 길이를 알아야함
 2. 문자열의 길이를 이용하여 문자열의 앞과 뒤를 검사해서 같은지를 확인
 3. 반복문을 모두 수행하지 않았어도, 중간에 같지 않은 문자가 발견된다면, 반복을 중단하고 회문이 아니라는 것을 출력

Questions(3)

```
#include <stdio.h>

//문자열이 회문인지 아닌지를 알려주는 함수
void pal(char* ptr, int len){
    int count = 0;
    for (int i = 0; i < len / 2; i++){
        //i번째 맨 처음과 맨 마지막 문자가 같으면 count값 증가
        if (ptr[i] == ptr[len - i - 1])
            count++;
        //같지 않으면 반복문 탈출
        else{
            printf("회문이 아닙니다.");
            break;
        }
    }
    //count값이 len/2와 같아야 회문
    if (count == len / 2)
        printf("회문입니다.");
}
```

```
//문자열의 길이를 알려주는 함수
int leng(char* ptr){
    int leng = 0;

    //null문자를 만날때까지 반복
    while (ptr[leng] != '\0')
        leng++;

    return leng;
}

int main(void){
    char str[100];
    int length;
    printf("문자열 입력 : ");
    scanf_s("%s", str, 100);
    length = leng(str); //문자열의 길이를 알기 위해 leng함수 호출
    pal(str, length); //문자열과 길이를 인자로 pal 함수에 전달

    printf("\n");
    return 0;
}
```


Contents

- Example (50 min)
- Questions (60 min)
- **Final Term (40min)**

Final Term(1)

다음은 정수형 배열 data[]를 오름차순으로 정렬하는 프로그램이다. 빈 칸을 채우시오

```
int i, j, temp;
int data[10] = { 9, 8, 2, 1, 4, 5, 6, 3, 7, 10 };
for ( i = 0; i < 10 ; i++) {
    for ( _____ ) { /* 변수i, j를 이용 */
        if ( _____ ) { /* 배열 data를 이용 */
            temp = data[j-1];
            data[j-1] = data[j];
            data[j] = temp;
        }
    }
}
```

Final Term(1)

다음은 정수형 배열 data[]를 오름차순으로 정렬하는 프로그램이다. 빈 칸을 채우시오

```
int i, j, temp;
int data[10] = { 9, 8, 2, 1, 4, 5, 6, 3, 7, 10 };
for ( i = 0; i < 10 ; i++) {
    for ( j = 9; j > i ; j-- ) { /* 변수 i, j를 이용 */
        if ( data[j-1] > data[j] ) { /* 배열 data를 이용 */
            temp = data[j-1];
            data[j-1] = data[j];
            data[j] = temp;
        }
    }
}
```

Final Term(2)

다음은 변수의 값을 서로 바꾸기 위해 만들어진 포인터 변수를 매개로 한 swap 함수이다.
프로그램의 문제점과 해결방법을 서술하시오

```
#include <stdio.h>
void pswap(int *p1, int *p2);
int main(void)
{
    int A=10, B=20;
    int *pA, *pB;
    pA=&A, pB=&B;
    pswap(pA, pB);

    return 0;
}
```

```
void pswap(int *p1, int *p2)
{
    int *temp;
    temp=p1;
    p1=p2;
    p2=temp;
}
```

Final Term(2)

다음은 변수의 값을 서로 바꾸기 위해 만들어진 포인터 변수를 매개로 한 swap 함수이다.
프로그램의 문제점과 해결방법을 서술하시오

문제점 및 해결방안

psswap 함수내의 포인터에 참조 연산자 '*'가 빠져있어 값의 교환이 일어나지 않는다.

따라서 다음과 같이 참조연산자 '*'를 넣어 고쳐야 한다.

```
*temp=*p1;
```

```
*p1=*p2;
```

```
*p2=*temp;
```

Final Term(3)

다음은 더블 포인터 입장에서의 swap 함수 이다. 빈칸을 채우시오.

```
#include <stdio.h>
void pswap(int **p1, int **p2);
int main(void) {
    int A=10, B=20;
    int *pA, *pB;
    pA=&A, pB=&B;
    pswap(_____);
    //함수 호출 후
    printf("pA가 가리키는 변수 : %d \n", _____);
    printf("pB가 가리키는 변수 : %d \n", _____ );
    return 0;
}
void pswap(int **p1, int **p2) {
    int *temp;
    _____;
    _____;
    _____;
}
```

Final Term(3)

다음은 더블 포인터 입장에서의 swap 함수 이다. 빈칸을 채우시오.

```
#include <stdio.h>
void pswap(int **p1, int **p2);
int main(void) {
    int A=10, B=20;
    int *pA, *pB;
    pA=&A, pB=&B;
    pswap(&pA, &pB);
    //함수 호출 후
    printf("pA가 가리키는 변수 : %d \n", *pA );
    printf("pB가 가리키는 변수 : %d \n", *pB );
    return 0;
}
void pswap(int **p1, int **p2) {
    int *temp;
    temp=*p1;
    *p1=*p2;
    *p2=temp;
}
```

Final Term(4)

프로그램 내에서 학번을 입력하고 바로 이름을 입력하려고 할 때 다음 출력결과와 같은 문제가 발생하였다. 이 문제가 발생한 원인과 해결방안을 서술하시오.

```
int num;  
char name[20];  
printf("학번을 입력하세요 : ");  
scanf("%d", &num);  
printf("이름을 입력하세요 : ");  
fgets(name, sizeof(name), stdin);
```

출력결과

학번을 입력하세요 : 315
이름을 입력하세요 : 학번 : 315
이름 :

Final Term(4)

프로그램 내에서 학번을 입력하고 바로 이름을 입력하려고 할 때 다음 출력결과와 같은 문제가 발생하였다. 이 문제가 발생한 원인과 해결방안을 서술하시오.

문제점 및 해결방안

scanf함수가 학번을 입력 받은 후에 버퍼에 남겨진 새줄 문자를 다음에 호출되는 fgets함수가 데이터로 받아들이기 때문이다 .

fflush(stdin); 함수를 사용하여 scanf함수와 gets함수가 공유하는 표준 입력 스트림버퍼를 비운다.

Q n A

Thanks