문항 선택

1 2 3 4 5
6 7

검토 완료

강의실 홈 —

강의정보 ▴
강의계획서

성적/출석관리 ▴
온라인출석부
오프라인출석부
성적부

수강생 알림 ▴
쪽지 보내기

기타 관리 ▾

학습활동 +

기초컴퓨터프로그래밍 **[Final-B:90]**

시작 일시	2018-12-17, 20:46:46
진행 상황	종료됨
완료 일시	2018-12-17, 21:52:02
소요시간	1 시간 5 분
성적	최고 90.00점 중 **75.00**점 (**83**%)

문제 **1**
풀이 완료
총 10.00 점에서
10.00 점 할당

Complete the reverse_array() function in the source code below properly so that we can get the following execution result.

Submit only the source code of your reverse_array() function.

Execution Result

```
1.10, 3.30, 5.50, 2.20, 6.60, 4.40
After Reverse
4.40, 6.60, 2.20, 5.50, 3.30, 1.10
```

```
#include <stdio.h>
void print_array(int n, double da[]) {
    int i;
    for (i=0; i<n-1; i++)
        printf("%6.2f, ", da[i]);
    printf("%6.2f\n", da[i]);
}

void reverse_array(int n, double da[]) {
    // Enter your code
}

int main(void) {
    double vec1[] = {1.1,3.3,5.5,2.2,6.6,4.4};

    print_array(sizeof(vec1)/sizeof(double), vec1);

    reverse_array(sizeof(vec1)/sizeof(double), vec1);

    printf("After Reverse\n");
    print_array(sizeof(vec1)/sizeof(double), vec1);

    return 0;
}
```

```
#include <stdio.h>
void print_array(int n, double da[]) {
    int i;
    for (i=0; i<n-1; i++)
                printf("%6.2f, ", da[i]);
    printf("%6.2f\n", da[i]);
}

void reverse_array(int n, double da[]) {
    int i;
```

문제 **2**
풀이 완료
총 10.00 점에서
10.00 점 할당

Complete the swap() and the main() functions in the source code below to swap values of two variables, a and b.

Execution Result

```
a: 10 and b: 20
After swap()
a: 20 and b: 10
```

```
#include <stdio.h>
void swap(// define parameters properly   )
{
}

int main(void)
{
    int a = 10, b = 20;

    printf("a: %d and b: %d\n", a, b);

    swap(// write parameters properly);
    printf("After swap()\n");
    printf("a: %d and b: %d\n", a, b);
    return 0;
}
```

```
#include <stdio.h>
void swap(int *a, int *b  )
{
        int temp;
        temp = *a;
        *a = *b;
        *b = temp;
}

int main(void)
```

문제 **3**

풀이 완료

총 15.00 점에서 15.00 점 할당

Complete the following program to get the following execution result.

You should complete two functions, distance1() and distance2(). both return the distance between two points.

- These two functions are different from each other in their parameter type.
- You can use any standard library functions declared in <math.h> such as sqrt() to compute distance.

Submit only your distance1() and distance2() functions for your answer.

Execution Result

```
The distance between (p1,p2) = 3.354102
The distance between (p1,p2) = 3.354102
```

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

typedef struct st_point {
        double x;
        double y;
} tpoint;


double distance1(tpoint p1, tpoint p2) {
    // Enter your code here
}

double distance2(tpoint *pp1, tpoint *pp2) {
    // Enter your code here
}

int main(void)
{
        tpoint p1 = {1.0, 3.0}, p2 = {-2.0, 4.5};

        printf("The distance between (p1,p2) = %lf\n", distance1(p1,p2));
        printf("The distance between (p1,p2) = %lf\n", distance2(&p1,&p2));

        return 0;
}
```

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

typedef struct st_point {
        double x;
        double y;
} tpoint;


double distance1(tpoint p1, tpoint p2) {
```

문제 **4**

풀이 완료

총 10.00 점에서 10.00 점 할당

Complete the following program to get the following execution result.

- point_swap() is a function that swaps values of two tpoint variables.
- Define parameters of point_swap() properly.
- Fill in [1] in main() properly which calls the point_swap().

Submit only your code for your answer.

Execution Result

```
P1 = <1.00,2.00>, P2 = <3.00,4.00>
P1 = <3.00,4.00>, P2 = <1.00,2.00>
```

```
#include <stdio.h>

typedef struct st_point {
        double x;
        double y;
} tpoint;


void point_swap([2 - define parameters properly]) {
    // Enter your code
}

int main(void)
{
        tpoint p1 = {1.0, 2.0}, p2 = {3.0, 4.0};

        printf("P1 = <%.2lf,%.2lf>, P2 = <%.2lf,%.2lf>\n",
                p1.x, p1.y, p2.x, p2.y);

        point_swap([1]);

        printf("P1 = <%.2lf,%.2lf>, P2 = <%.2lf,%.2lf>\n",
                p1.x, p1.y, p2.x, p2.y);

        return 0;
}
```

```
#include <stdio.h>

typedef struct st_point {
        double x;
        double y;
} tpoint;


void point_swap(tpoint *pp1, tpoint *pp2) {
    tpoint temp;
```

문제 **5**

풀이 완료

총 15.00 점에서 15.00 점 할당

Write a program that does the followings.

1. Get an integer number from user input. let it be "n"
2. Get integer numbers as many as "n" from user input
3. Print out those integer numbers **in the reverse order of the input**
4. You should use dynamic memory allocation to store integer numbers from user input
   That is, your program should use "malloc()" and "free()" .

For students who are hard to understand the above problem written in English.

- 입력 받을 정수의 개수와 이에 해당하는 개수만큼의 정수를 입력 받아 이를 역순으로 출력하는 프로그램을 작성하시오. 단 입력 받은 정수를 저장하는 과정에서 동적 메모리 할당 및 해제 함수인 malloc()과 free() 함수를 사용하시오.

Submit your source code.

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

int main(){
        int n,i;
        int *b;
        printf("n 값 : ");
        scanf("%d",&n);
        b = malloc(sizeof(int)*n);
```

문제 **6**

풀이 완료

총 15.00 점에서 15.00 점 할당

A palindromic number or numeral palindrome is a number that remains the same when its digits are reversed. Like 16461, for example, it is "symmetrical".

Complete the source code below determining whether a positive integer input by the user is a palindromic number or not by making your own is_palindromic_number() function.

```
#include <stdio.h>
int is_palindromic_number([1])
{
    // Enter your code
}

int main(void)
{
    unsigned int n;

    printf("Enter a positive integer : ");
    scanf("%u",&n);

    if (is_palindromic_number(n))
        printf("is a palindromic number\n");
    else
        printf("is not a palindromic number\n");

    return 0;
}
```

```
#include <stdio.h>
int is_palindromic_number(unsigned int n)
{
    unsigned int reverse_n=0;
    unsigned int i;
    for(i=n;i!=0;i=i/10){
        reverse_n = reverse_n*10;
                reverse_n += i%10;
        }
        if(reverse_n==n){
```

문제 **7**

답하지 않음

총 15.00 점

Complete hexa2decimal() in the source code below to get the following execution result.

The hexa2decimal() return a decimal integer value corresponding to a hexa string given by the parameter "phexa".

You can safely assume that all hexa string starts with "0x" and only 0~9 and A~F (Capital/Upper) characters are used.

Submit your hexa2decimal().

Execution Result

```
0x1F34 = 7988
0x34CD56 = 3460438
```

```
#include <stdio.h>
int hexa2decimal(char *phexa) {
         // Enter your code
}
int main(void) {
        char hexa1[] = "0x1F34";
        char hexa2[] = "0x34CD56";

        printf("%s = %d\n", hexa1, hexa2decimal(hexa1));
        printf("%s = %d\n", hexa2, hexa2decimal(hexa2));

        return 0;
}
```