• 문항 선택
1 2 3 4
5 6 7 8
9 10
한 문제씩 검토
검토 완료
강의실 홈
강의정보
· 교수계획표 (국문)
· 교수계획표 (영문)
성적/출석관리
· 동영상이수현황
· 스마트출석부
· 성적부
수강생 알림
· 쪽지 보내기
기타 관리
학습활동

시작 일시	2022-12-03 21:52
진행 상황	종료됨
완료 일시	2022-12-07 16:38
소요시간	3 일 18 시간
성적	최고 16.00점 중 **15.00**점 (94%)

문제 **1**
정답
총 2.00 점에서 2.00 점 할당
문제 표시

Complete the source code below by choosing a proper expression for the blank.

This program allocates memory dynamically to store random numbers using malloc().

The number of random numbers is provided by the user through the variable "nsize".

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

int main(void)
{
    int* na;
    unsigned int nsize;
    int i;

    puts("Enter N (>0):");
    scanf("%u", &nsize);
    srand(time(0));

    na = malloc( [sizeof(int)*nsize] ✔ ;
    if (na) {
        for (i=0; i<nsize; i++) {
            na[i] = rand()%10000;
            printf("%d ", na[i]);
        }
        free(na);
     }

     return 0;
}
```

문제 **2**
정답
총 1.00 점에서 1.00 점 할당
문제 표시

Complete the source code below by filling in the blank with a proper C standard function/macro.

The macro/function is used to check if its argument (which must have a scalar type) compares equal to zero. If it does, it outputs implementation-specific diagnostic information on the standard error output and calls abort().

In the following example code, it is called to check the result of malloc().

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

int main(void)
{
        int na[100], i;
        int *p_nacopy = NULL;

        for (i=0; i<100; i++)
                na[i] = i;

        p_nacopy = malloc(sizeof(na));
        [assert] ✔ (p_nacopy);

        free(p_nacopy);

        return 0;
}
```

문제 **3**
정답
총 3.00 점에서 3.00 점 할당
문제 표시

For the following variables and memory blocks used in the source code below, classify their lifetime.

Variable	Declared Line	Lifetime
g_sum	line 5	Static ✔
num	line 7	Automatic ✔
num	line 11	Automatic ✔
sum	line 12	Static ✔
ns	line 18	Automatic ✔
pns_copy	line 19	Automatic ✔
memory block pointed by pns_copy	line 19	Dynamic ✔
i	line 20	Automatic ✔
sum	line 20	Automatic ✔

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <assert.h>
4
5  int g_sum = 0;
6
7  int add(int num) {
8          g_sum += num;
9          return g_sum;
10 }
11 int add2(int num) {
12         static int sum = 0;
13         sum += num;
14         return sum;
15 }
16
17 int main(void) {
18         int ns[] = {1, 2, 3, 4, 5};
19         int *pns_copy = malloc(sizeof(ns));
20         int i, sum;
21
22         assert(pns_copy);
23         for (i=0; i<5; i++) {
24                 pns_copy[i] = ns[i];
25                 sum = add(ns[i]);
26         }
27         printf("sum of ns : %d\n", sum);
28
29         for (i=0; i<5; i++) {
30                 sum = add2(pns_copy[i]);
31         }
32         printf("sum of pns_copy : %d\n", sum);
33
34         free(pns_copy);
35
36         return 0;
37 }
```

문제 **4**
정답
총 1.00 점에서 1.00 점 할당
문제 표시

In the source code below, fill in the blank with a proper C standard library function.

We want that the function copy all values in the array na[100] to the memory block pointed by p_nacopy.

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

int main(void)
{
        int na[100], i;
        int *p_nacopy = NULL;

        for (i=0; i<100; i++)
                na[i] = i;

        p_nacopy = malloc(sizeof(na));

         /* for block
         for (i=0; i<100; i++)
             p_nacopy[i] = na[i];
          */
         // In the next statement, we want to do the same work with
         // the above "for block" if it is not commented.
         [memcpy] ✔ (p_nacopy, na, sizeof(na));

        free(p_nacopy);

        return 0;
}
```

문제 **5**
정답
총 2.00 점에서 2.00 점 할당
문제 표시

You are given a text file named "input.txt", and its contents are shown below.

input.txt

```
1.1
 2.2 3.3 4.4 end
```

Choose the proper codes for the source code below to get the following execution result.

Execution Result

```
1.1
 2.2 3.3 4.4 end
1.1
 2.2 3.3 4.4 end
1.100000
2.200000
3.300000
4.400000


--------------------------------
Process exited after 0.056 seconds with return value 0
Press any key to continue . . .
```

```
#include <stdio.h>
#include <assert.h>
int main(void)
{
    FILE* ifp;
    int c;
    char s[1024];
    double v;

    ifp = fopen("input.txt", "r");
    assert(ifp!=NULL);
    while( (c=fgetc(ifp)) [!= EOF] ✔ )
        fputc(c, stdout); // == putchar(c);
    fprintf(stdout,"\n");
    fclose(ifp);

    ifp = fopen("input.txt", "r");
    assert(ifp!=NULL);
    while( fgets(s,sizeof(s),ifp) [!=NULL] ✔ )
        fputs(s, stdout); // == puts(s);
    fprintf(stdout,"\n");
    fclose(ifp);

    ifp = fopen("input.txt", "r");
    assert(ifp!=NULL);
    while( fscanf(ifp,"%lf",&v) [==1] ✔ )
        fprintf(stdout,"%f\n", v); // == printf("%f\n", v);
    fprintf(stdout,"\n");
    fclose(ifp);
    return 0;
}
```

문제 **6**
부분적으로 맞음
총 2.00 점에서 1.00 점 할당
문제 표시

Write the data stored in two files (text.txt, binary.bin) generated by the execution of the following program in hexadecimal.

test.txt

0x [39] ✘ [32] ✘ [30] ✘ [31] ✘

binary.bin

0x [05] ✔ [04] ✔ [00] ✔ [00] ✔

```
#include <stdio.h>

// The CPU running this program is Intel Core i5.

int main(void)
{
    const char* txt_file = "text.txt";
    const char* bin_file = "binary.bin";
    FILE* fp;
    int data = 1029;

    fp = fopen(txt_file,"w");
    if(fp==NULL) {
        printf("%s cannot be opened.\n", txt_file);
        return -1;
    }
    fprintf(fp,"%d", data);
    fclose(fp);

    fp = fopen(bin_file,"wb");
    if(fp==NULL) {
        printf("%s cannot be opened.\n", bin_file);
        return -2;
    }
    fwrite(&data,sizeof(int),1,fp);
    fclose(fp);
    return 0;
}
```

문제 **7**
정답
총 1.00 점에서 1.00 점 할당
문제 표시

You are given a text file named "input.txt", and its contents are as follows.

input.txt

```
1.1
 2.2 3.3 4.4
```

Using the "input.txt", we want to get the same following execution results from the two source codes below.

Choose a proper code for the second source code.

Execution Result

```
1.100000
2.200000
3.300000
4.400000


--------------------------------
Process exited after 0.05128 seconds with return value 0
Press any key to continue . . .
```

```
// Source Code 1
#include <stdio.h>
#include <assert.h>

int main(void)
{
    FILE* ifp;
    double v;

    ifp = fopen("input.txt", "r");
    assert(ifp!=NULL);
    while( fscanf(ifp,"%lf",&v) != EOF )
        fprintf(stdout,"%f\n", v);
    fprintf(stdout,"\n");
    fclose(ifp);
    return 0;
}
```

```
// Source Code 2
#include <stdio.h>
#include <assert.h>

int main(void)
{
    FILE* ifp;
    double v;

    ifp = fopen("input.txt", "r");
    assert(ifp!=NULL);
    while(1) {
        fscanf(ifp,"%lf",&v);
        [if( feof(ifp) ) break;] ✔
        fprintf(stdout,"%f\n", v);
    }
    fprintf(stdout,"\n");
    fclose(ifp);
    return 0;
}
```

문제 **8**
정답
총 1.00 점에서 1.00 점 할당
문제 표시

What is the name of the function that takes command line arguments? [main] ✔

Actually, this function must exist and be unique in every C program source code that runs independently.

문제 **9**
정답
총 1.00 점에서 1.00 점 할당
문제 표시

What will be the output of the following program, assuming that this program is executed in the command line as follows:

C:\mypgr hello 123

Execution Result

[mypgr,hello,123,] ✔

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
        int i;
        for (i=0 ; i<argc ; i++)
                printf("%s,", argv[i]);
        return 0;
}
```

문제 **10**
정답
총 2.00 점에서 2.00 점 할당
문제 표시

Write the names of the C library functions explained below. These functions are defined in <stdlib.h>

- [atoi] ✔ : the function that converts/interprets a character string into an integer value
- [atof] ✔ : the function that converts/interprets a character string into a floating-point value.

검토 완료