

12-1.c

The image shows a Visual Studio Code editor window with the following components:

- Editor Window:** Displays the source code of `12-1.c`. The code defines two functions, `get_numbers` and `print_numbers`, and a `main` function. The `main` function uses `scanf` to read integers into an array, then calls `get_numbers` and `print_numbers` to process and display the data. Comments in Korean explain the code's logic.
- Output Window:** Located at the bottom, it shows the compilation output. It indicates that the build was successful (1 success, 0 failures) and provides the file path for the executable: `C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터 프로그래밍 및 실습 (F024-1)\Visual Studio\12주차 실습\1212\12-1.exe`.
- Solution Explorer:** Located on the right, it shows the project structure. The file `12-1.c` is highlighted under the `1212` project.

The code in `12-1.c` is as follows:

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

void get_numbers(int* numbers, int n);           // 함수 정의
void print_numbers(int* numbers, int n);         // 함수 정의

int main(void) {
    int count;
    int* numbers;
    printf("정수의 개수: ");
    scanf("%d", &count);                        // 정수의 개수 입력
    numbers = (int*)malloc(count * sizeof(int)); // count만큼 동적 메모리 할당
    if (numbers == NULL) {
        printf("동적 메모리 할당 오류");        // 에러 메시지 출력
        exit(1);
    }

    get_numbers(numbers, count);                // get_numbers 함수 실행
    print_numbers(numbers, count);              // print_numbers 함수 실행
    free(numbers);                             // 할당된 메모리 반환
    return 0;
}

void get_numbers(int* numbers, int n) {
    for (int i = 0; i < n; i++) {
        printf("정수를 입력하시오: ");
        scanf("%d", &numbers[i]);              // n개 만큼 정수 입력
    }
}

void print_numbers(int* numbers, int n) {
    int sum = 0;
    printf("### 입력된 정수: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", numbers[i]);             // 입력된 정수 출력
        sum += numbers[i];                     // 출력마다 출력되는 정수를 sum 변수에 더함
    }
    printf("\n### 입력된 정수들의 합: %d", sum); // 입력된 정수들의 합 sum 출력
}
```

12-1.c

The image shows a Visual Studio IDE with a C program named 12-1.c and its execution output.

Source Code (12-1.c):

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void get_numbers(int* numbers, int n);           // 함수 정의
6  void print_numbers(int* numbers, int n);         // 함수 정의
7
8  int main(void) {
9      int count;
10     int* numbers;
11     printf("정수의 개수: ");
12     scanf("%d", &count);                         // 정수의 개수 입력
13     numbers = (int*)malloc(count * sizeof(int));   // count만큼 동적 메모리 할당
14     if (numbers == NULL) {
15         printf("동적 메모리 할당 오류");          // 에러 메시지 출력
16         exit(1);
17     }
18
19     get_numbers(numbers, count);                  // get_numbers 함수 실행
20     print_numbers(numbers, count);                // print_numbers 함수 실행
21     free(numbers);                                // 할당된 메모리 반환
22     return 0;
23 }
24
25 void get_numbers(int* numbers, int n) {
26     for (int i = 0; i < n; i++) {
27         printf("정수를 입력하시오: ");
28         scanf("%d", &numbers[i]);                // n개 만큼 정수 입력
29     }
30 }
31
32 void print_numbers(int* numbers, int n) {
33     int sum = 0;
34     printf("### 입력된 정수: ");
35     for (int i = 0; i < n; i++) {
36         printf("%d ", numbers[i]);                // 입력된 정수 출력
37         sum += numbers[i];                        // 출력마다 출력되는 정수를 sum 변수에 더함
38     }
39     printf("\n### 입력된 정수들의 합: %d", sum);    // 입력된 정수들의 합 sum 출력
40 }
```

Output Window:

```
정수의 개수: 5
정수를 입력하시오: 20
정수를 입력하시오: -5
정수를 입력하시오: 15
정수를 입력하시오: 50
정수를 입력하시오: -30
### 입력된 정수: 20 -5 15 50 -30
### 입력된 정수들의 합: 50
C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12주
1212\x64\Debug\1212.exe(프로세스 22140개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

Output Window (Build):

```
출력
출력 보기 선택(S): 빌드
```

Taskbar:

12:21:29 2023-12-14

12-2.c

The image shows a Visual Studio Code editor window with a C program named 12-2.c. The program defines two functions: `get_strings` and `print_strings`, and a `main` function that uses them. The `main` function prompts the user for the number of strings (`n`), allocates memory for an array of strings, and then calls `get_strings` to fill the array. Finally, it calls `print_strings` to display the strings.

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void get_strings(char** strings, int n);           // get_strings함수 정의
6  void print_strings(char** strings, int n);         // print_strings함수 정의
7
8  int main(void) {
9      int n;
10     char** strings;
11     printf("문자열의 개수: ");
12     scanf("%d", &n);                               // 문자열의 개수 입력
13     strings = (char **)malloc(n * sizeof(char*));   // n개 만큼 동적 메모리 할당
14     for (int i = 0; i < n; i++) {
15         strings[i] = (char *)malloc(100 * sizeof(char)); // 각각 100만큼 동적 메모리 할당
16         if (strings[i] == NULL) {
17             printf("동적 메모리 할당 오류");
18             exit(1);
19         }
20     }
21     get_strings(strings, n);                         // get_strings함수 실행
22     print_strings(strings, n);                       // print_strings함수 실행
23     free(strings);                                  // 할당된 메모리 반환
24     return 0;
25 }
26
27 void get_strings(char** strings, int n) {
28     for (int i = 0; i < n; i++) {
29         printf("문자열을 입력하시오: ");
30         scanf("%s", strings[i]);                    // n개 만큼 문자열 입력
31     }
32 }
33
34 void print_strings(char** strings, int n) {
35     printf("### 입력된 문자열:\n");                  // 헤딩 출력
36     for (int i = 0; i < n-1; i++) {
37         printf("W%sW", strings[i]);                  // 입력된 문자열 출력
38     }
39     printf("W%sW", strings[n-1]);
40 }
41
```

The output window shows the following text:

```
출력 보기 선택(S): 빌드
오후 1:09에 빌드를 시작함...
1>----- 빌드 시작: 프로젝트: 1212, 구성: Debug x64 -----
1>12-2.c
1>1212.vcxproj -> C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12주차_실습\1212\1212.vcxproj
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛰어 =====
===== 빌드이(가) 오후 1:09에 완료되었으며, 01.166 초이(가) 걸림 =====
```

The bottom status bar shows the temperature as 9°C and the time as 1:09:38 on 2023-12-14.

12-2.c

The image shows a screenshot of the Microsoft Visual Studio IDE. The main window displays a C program named `12-2.c`. The program defines two functions, `get_strings` and `print_strings`, and a `main` function. The `main` function prompts the user to enter the number of strings (`n`), then uses `get_strings` to read `n` strings into an array. It then uses `print_strings` to print the strings. The `get_strings` function uses `scanf` to read strings into an array of dynamically allocated memory. The `print_strings` function uses `printf` to print the strings.

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void get_strings(char** strings, int n);           // get_strings함수 정의
6  void print_strings(char** strings, int n);         // print_strings함수 정의
7
8  int main(void) {
9      int n;
10     char** strings;
11     printf("문자열의 개수: ");
12     scanf("%d", &n);                               // 문자열의 개수 입력
13     strings = (char**)malloc(n * sizeof(char*));     // n개 만큼 동적 메모리 할당
14     for (int i = 0; i < n; i++) {
15         strings[i] = (char*)malloc(100 * sizeof(char)); // 각각 100만큼 동적 메모리 할당
16         if (strings[i] == NULL) {
17             printf("동적 메모리 할당 오류");         // 에러 메시지 출력
18             exit(1);
19         }
20     }
21     get_strings(strings, n);                         // get_strings함수 실행
22     print_strings(strings, n);                       // print_strings함수 실행
23     free(strings);                                  // 할당된 메모리 반환
24     return 0;
25 }
26
27 void get_strings(char** strings, int n) {
28     for (int i = 0; i < n; i++) {
29         printf("문자열을 입력하시오: ");
30         scanf("%s", strings[i]);                   // n개 만큼 문자열 입력
31     }
32 }
33
34 void print_strings(char** strings, int n) {
35     printf("### 입력된 문자열:\n");                // 헤딩 출력
36     for (int i = 0; i < n-1; i++) {
37         printf("W%sW", " ", strings[i]);          // 입력된 문자열 출력
38     }
39     printf("W%sW", " ", strings[n-1]);
40 }
41
```

The Output window shows the execution results:

```
문자열의 개수: 5
문자열을 입력하시오: Tesla
문자열을 입력하시오: 현대
문자열을 입력하시오: Ford
문자열을 입력하시오: 기아
문자열을 입력하시오: BMW
### 입력된 문자열:
{"Tesla", "현대", "Ford", "기아", "BMW"}
C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12
1212\x64\Debug\1212.exe(프로세스 23396개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

The bottom status bar shows the file encoding as UTF-8 and the current line is 26, column 1.

12-3.c

The image shows a Visual Studio Code editor window with the following components:

- Editor:** Displays the source code of `12-3.c`. The code defines a `CONTACT` struct, `get_contacts` and `print_strings` functions, and a `main` function that prompts the user for the number of contacts and their details (name and email).
- Output Window:** Shows the output of the program. It indicates that the build was successful and the program executed without errors. The output text is:
오류 보기 선택(S): 빌드
오후 1:32에 빌드를 시작함...
>----- 빌드 시작: 프로젝트: 1212, 구성: Debug x64 -----
>12-3.c
>1212.vcxproj -> C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12주차_실습\1212\1212\Debug\1212.exe
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 1:32에 완료되었으며, 00.816 초이(가) 걸림 =====
|
- Explorer:** Shows the file structure of the project, including `12-1.c`, `12-2.c`, and `12-3.c`.

12-3.c

The image shows a Microsoft Visual Studio IDE with a C program named 12-3.c. The program is designed to manage a list of contacts. It defines a `CONTACT` struct with `name` and `email` fields. The `main` function prompts the user for the number of contacts, dynamically allocates memory for an array of `CONTACT` structs, and then prompts for each contact's name and email. The `get_contacts` function handles the input, and the `print_strings` function displays the resulting list of contacts in a formatted table.

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  typedef struct cont {           // 구조체 선언
6      char name[100];
7      char email[100];
8  }CONTACT;
9
10 void get_contacts(CONTACT* conts, int n); // 함수 정의
11 void print_strings(CONTACT* conts, int n); // 함수 정의
12
13 int main(void) {
14     CONTACT* conts;
15     int n;
16     printf("주소의 개수: ");
17     scanf("%d", &n);           // 주소의 개수 입력
18     conts = (CONTACT*)malloc(n * sizeof(CONTACT)); // n만큼 동적 메모리 할당
19     if (conts == NULL) {
20         printf("동적 메모리 할당 오류"); // 에러 메시지 출력
21         exit(1);
22     }
23     get_contacts(conts, n);     // get_contacts 함수 실행
24     print_strings(conts, n);    // print_strings 함수 실행
25     free(conts);               // 메모리 반환
26     return 0;
27 }
28
29 void get_contacts(CONTACT* conts, int n) {
30     for (int i = 0; i < n; i++) {
31         printf("이름을 입력하시오: ");
32         scanf("%s", conts[i].name); // 이름 입력
33         printf("이메일을 입력하시오: ");
34         scanf("%s", conts[i].email); // 이메일 입력
35     }
36 }
37
38 void print_strings(CONTACT* conts, int n) {
39     printf("=====Wn");
40     printf("이름WtWt이메일Wn");
41     printf("=====Wn"); // 헤딩 출력
42     for (int i = 0; i < n; i++) {
43         printf("%sWtWt%sWn", conts[i].name, conts[i].email); // 각 이름 이메일 출력
44     }
45     printf("=====");
46 }
```

The output window shows the program's execution. It prompts for the number of contacts (3), then for each contact's name and email. The resulting list of contacts is displayed in a formatted table.

```
주소의 개수: 3
이름을 입력하시오: 김야구
이메일을 입력하시오: baseball@gmail.com
이름을 입력하시오: 박축구
이메일을 입력하시오: soccer@hotmail.com
이름을 입력하시오: 이배구
이메일을 입력하시오: volleyball@naver.com
=====
이름      이메일
=====
김야구    baseball@gmail.com
박축구    soccer@hotmail.com
이배구    volleyball@naver.com
=====
C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12주차 실
1212\x64\Debug\1212.exe(프로세스 19780개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

12-4.c

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

typedef struct node {
    int data;
    struct node* link;
} NODE;

int get_integer();           // 함수 정의
void print_list(NODE* plist); // 함수 정의
void free_list(NODE* plist); // 함수 정의

int main() {
    NODE* plist = NULL;

    while (1) {
        int n = get_integer();
        if (n < 0) break;           // n이 음수면 종료
        NODE* node = (NODE*)malloc(sizeof(NODE)); // 동적 메모리 할당
        if (node == NULL) {
            printf("메모리 할당 오류\n");
            exit(1);
        }
        node->data = n;           // node 연결
        node->link = NULL;
        if (plist == NULL) {
            plist = node;
        }
        else {
            NODE* temp = plist;
            while (temp->link != NULL) {
                temp = temp->link;
            }
            temp->link = node;
        }

        print_list(plist);         // print_list 함수 실행
        free_list(plist);          // free_list 함수 실행
    }

    return 0;
}
```

```
int get_integer() {
    int n;
    printf("양의 정수를 입력하십시오 (종료하려면 -1을 입력): ");
    scanf("%d", &n);           // 정수 입력
    return n;                  // 정수 반환
}

void print_list(NODE* plist) {
    printf("### 입력된 연결 리스트: ");           // 헤딩 출력
    NODE* temp = plist;
    while (temp != NULL) {
        printf("%d", temp->data);           // 데이터 출력
        if (temp->link != NULL) {
            printf("->");
        }
        else {
            printf("->NULL\n");
        }
        temp = temp->link;           // 다음 링크로 이동
    }
}

void free_list(NODE* plist) {
    NODE* temp;
    while (plist != NULL) {
        temp = plist;
        plist = plist->link;           // 다음 링크로 이동
        free(temp);                   // 메모리 반환
    }
}
```

% 0 1 ↑ ↓ | ↺ ↻ ⌂ ⌚

빌드

12-3.c
12-4.c
코드를 생성하고 있습니다...

1212.vcxproj -> C:\Users\Wchany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\WVisual Studio\12주차 실습\1212\64\Debug\1

===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====

===== 빌드이(가) 오후 1:57에 완료되었으며, 02.270 초이(가) 걸림 =====

12-4.c

The image shows a screenshot of the Microsoft Visual Studio IDE. The main window displays a C program named `12-4.c` which implements a linked list. The code includes headers for `<stdio.h>` and `<stdlib.h>`, defines a `node` structure with an integer `data` and a pointer to the next `node`, and implements functions `get_integer`, `print_list`, and `free_list`. The `main` function uses a `while` loop to read integers from the user and build the linked list until a `-1` is entered. It then prints the list and frees the memory.

The Output window shows the execution results, indicating that the program successfully processed 73 inputs and terminated with a return code of 0. The input sequence is: 20, 10, 50, 35, 77, 42, 66, -1.

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

typedef struct node {
    int data;
    struct node* link;
} NODE;

int get_integer();           // 함수 정의
void print_list(NODE* plist); // 함수 정의
void free_list(NODE* plist); // 함수 정의

int main() {
    NODE* plist = NULL;

    while (1) {
        int n = get_integer();
        if (n < 0) break;           // n이 음수면 종료
        NODE* node = (NODE*)malloc(sizeof(NODE)); // 동적 메모리 할당
        if (node == NULL) {
            printf("메모리 할당 오류\n");
            exit(1);
        }
        node->data = n;              // node 연결
        node->link = NULL;
        if (plist == NULL) {
            plist = node;
        } else {
            NODE* temp = plist;
            while (temp->link != NULL) {
                temp = temp->link;
            }
            temp->link = node;
        }
    }

    print_list(plist);              // print_list 함수 실행
    free_list(plist);               // free_list 함수 실행
    return 0;
}

int get_integer() {
    int n;
    printf("양의 정수를 입력하십시오 (종료하려면 -1을 입력): ");
}
```

Microsoft Visual Studio 디버그

양의 정수를 입력하십시오 (종료하려면 -1을 입력): 20
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 10
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 50
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 35
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 77
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 42
양의 정수를 입력하십시오 (종료하려면 -1을 입력): 66
양의 정수를 입력하십시오 (종료하려면 -1을 입력): -1
입력된 연결 리스트: 20->10->50->35->77->42->66->NULL

C:\Users\chany\OneDrive - Ajou University\Mathematics\2023-2-2\컴퓨터프로그래밍및실습 (F024-1)\Visual Studio\12주차 실
1212\x64\Debug\1212.exe(프로세스 7492개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

출력 보기 선택(S): 빌드

호출 계층 구조 오류 목록 출력 책갈피

빌드 성공

9°C 흐림

검색

오후 1:57:58
2023-12-14