

컴퓨터공학 설계실험 1 중간고사 (2008년 4월 26일)

총점: 100점

시간: 75분

주의 사항

1. 시험 시작 전에 페이지를 넘기지 말 것.
2. 제출하는 모든 시험 답안지에 이름, 학번, 소속 반을 반드시 쓸 것.
3. “끝” 하는 말에 모든 행동을 정지 할 것.

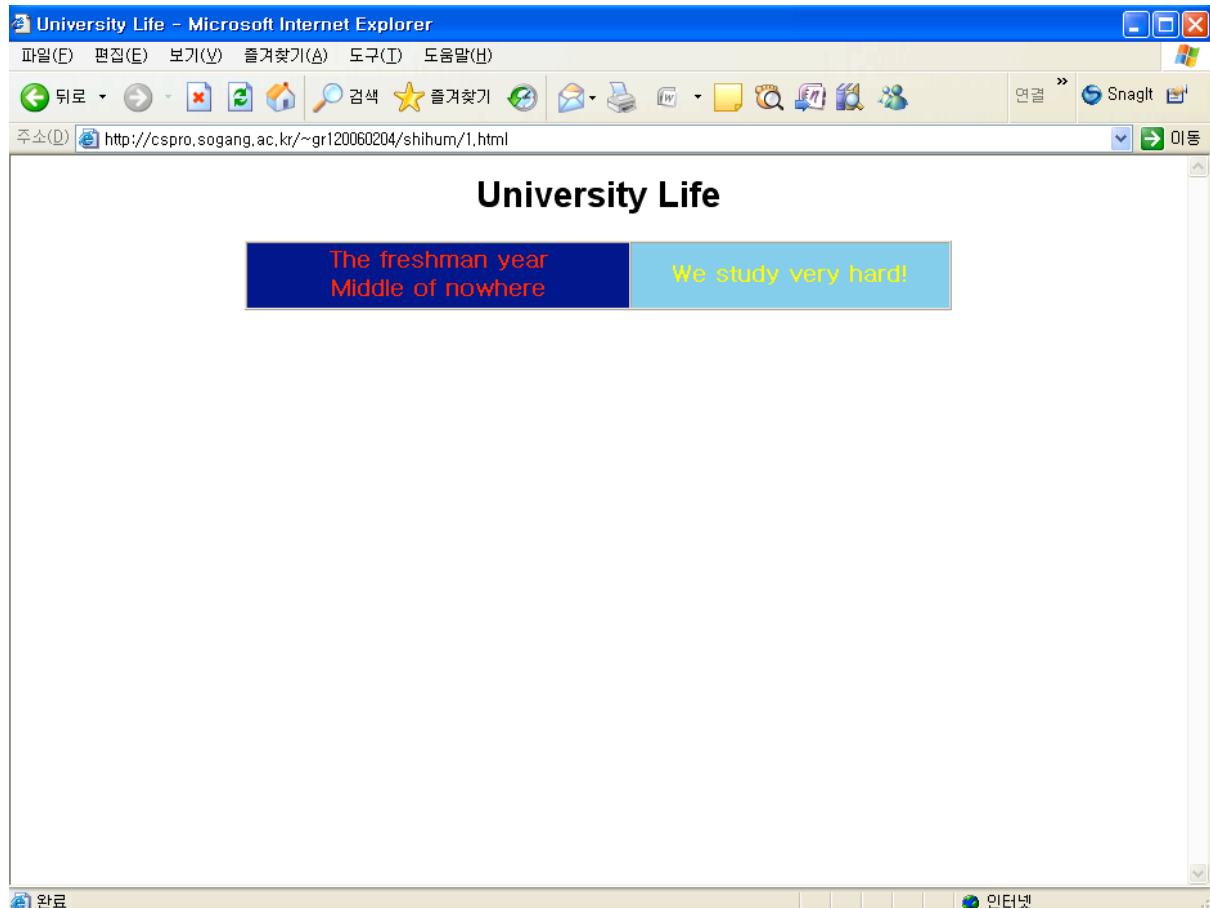
Good Luck!

[이름]:
[학번]:
[소속 반]: 반

1. [8 점] 아래의 <기본조건>에 따라 HTML 문서를 작성하시오.

<기본조건>

- title 은 'University Life'로 한다.
- 제목인 University Life 는 가운데 정렬에 글자크기는 5 이며, Arial 체에 굵은 글씨로 한다
- 전체 표의 너비는 550 픽셀이며, 테두리의 굵기는 1 로 하며, 셀 안쪽 여백은 3, 셀과 셀 사이의 간격은 0 으로 한다.
- "The freshman year.." 셀의 너비는 300 픽셀, 높이는 50 픽셀이며, "We study very hard!" 셀의 전체 너비는 250 픽셀이다
- The freshman year.., We study very hard 의 셀 배경색은 각각 darkblue 와 skyblue 이다.
- 글자 크기는 보통으로 하며 "The freshman year"는 빨간색 굵은 글씨체에 가운데 정렬이며 "We study very hard"는 노란색 굵은 글씨체에 가운데 정렬로 한다.
- 기타 조건이 주어지지 않은 것은 아래 화면과 같이 보이도록 한다.



답 :

2. [총 9 점] 다음 물음에 답하십시오.

[각 1 점] 빈칸에 적당한 기호를 쓰시오.

- PHP 에서 주석문을 사용할 때에, 한 줄만을 주석처리 하고 싶을 때는 1) _____를 사용하면 되고, 한 줄 이상의 범위를 주석처리 하고 싶을 때는 2) _____로 시작해서 3) _____로 닫아주면 된다.

[각 2 점] 다음 문장의 참, 거짓을 판단하십시오.

- 4) PHP 프로그래밍에서 변수명의 길이에 제한은 없다. (T / F)
- 5) C 언어와 같이 변수 타입은 미리 선언해주어야 한다. (T / F)
- 6) 윈도우즈에서 <? ... ?> 대신 <% ... %>를 사용할 수 있다. (T / F)

3. [총 10 점] 아래 소스 코드를 보고 물음에 답하여라.

```

<?
$count = 10;
if(!$_COOKIE['ip'])
{
setcookie( "ip", $_SERVER['REOMTE_ADDR']
$count++;
}
echo( $count );
?>

```

- 1) [2 점] 이 프로그램을 웹 브라우저로 봤을 때 처음 화면에 출력되는 값은? ____
- 2) [2 점] F5 나 refresh 를 했을 때 출력되는 값은? _____
- 3) [6 점] 이렇게 작동하는 이유는 무엇인가? (if 의 조건부는 무엇을 의미하는지)

4. [총8점/각2점] 다음의 내용이 참인지 거짓인지 (T / F)로 판단하시오.

- 1) VS 6.0은 에디터, 컴파일러, 디버거의 기능을 모두 포함한 IDE 환경을 제공한다.
(T / F)
- 2) Breakpoint를 설정하지 않았을 경우 소스를 빌드한 후 단축키 F5를 누르거나, 단축키 Ctrl+F5를 누르거나 같은 작업을 실행하게 된다. (T / F)
- 3) 시간의 측정을 위해 사용하는 clock() 함수는 현재 시각을 초 단위로 반환한다.
(T / F)
- 4) VS 6.0은 링킹 과정을 배제하고 컴파일 과정만을 따로 수행할 수 있다. (T / F)

5. [총10점/각2점] VS 6.0의 디버그 모드에서 작동하는 다음 단축키들이 수행하는 기능에 대해 기술하시오.

- 1) F9
- 2) F10
- 3) F11
- 4) Shift + F9

5) Shift + F5

6.[총8점/각4점] 다음의 소스는 최대 공통 문자를 출력하기 위해 작성된 프로그램의 일부이다. 다음의 소스를 바탕으로 문제에 답하시오

```

src_2_3_20040971 - Microsoft Visual C++ - [main.cpp *]
File Edit View Insert Project Build Tools Window Help
[Globals] [All global members] main
Workspace 'src_2_3_20040971'
  src_2_3_20040971
    Source File
      main.cpp
      util.cpp
    Header File
      my.h
    Resource File

void main(void)
{
    clock_t start, finish;
    double duration;

    char ch, str[LINE_MAX][STR_MAX];
    int i=0, j=0, k=0;

    printf("input small letter (alphabet)\nTo complete -- PRESS ESC KEY --\n");

    while(1)
    {
        // 키 입력 알고리즘. 소문자 알파벳과 백스페이스, 리턴, ESC를 제외하고는 입력되지 않음
        ch=getch();

        if(i<STR_MAX && ch>=97&&ch<=122)
        {
            str[j][i]=ch;
            i++;
            printf("%c",ch);
        }
        else if(ch==13) // 엔터가 눌렸을때 값 입력완료
        {
            str[j][i]='\0';
            j++;
            i=0;
            printf("\n");
            printf("%d. ",j+1);
        }
        else if(ch==27 || j>=LINE_MAX) //ESC가 눌러질 경우 루틴 종료
        {
            str[j][i]='\0';
            break;
        }
        else
            printf("Wa");
    }

    printf("\nResult... \n");

    for(k=0;k<j;k+=2)
    {
        initArray(); //배열 초기화
        setArray(0,str[k]); //NA[]와
        setArray(1,str[k+1]); //NB[]에 값입력
        printf("%d. and %d. = ",k,k+1);
        resultPrint(); // 결과 출력
    }

    duration=((double)(finish-start)) / CLK_TCK;
    printf("Elapsed Time : %.2f(sec)\n",duration); // 수행시간 출력
}

```

위 소스에서 프로그램의 수행 시간을 측정하기 위해서 아래의 구문을 삽입하려고 한다.

- (1) start=clock();
- (2) finish=clock();

사용자의 입력에 필요한 시간을 배제하고 순수 연산에 필요한 수행 시간만을 측정하기 위해서 (1), (2)의 구문을 올바른 위치에 삽입하시오. (위 그림에 직접 표시)

7. [10 점] 다음의 UNIX Shell 프로그램을 작성하시오.

view 프로그램은 file 이름을 인자로 받아 해당 file 의 내용을 화면에 출력한다. 만약 file 이 존재하지 않으면 화면에 'file not found' 를 출력한다. 만약 인자를 입력하지 않으면 화면에 'USAGE:view filename'를 출력한다.

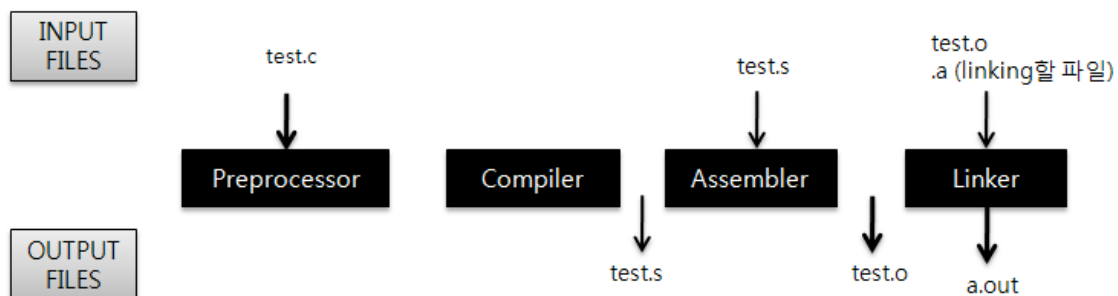
<프로그램 수행 예>

```
$ ./view
USAGE:view filename
$ ./view asdf
file not found
$ cat test
contents...
$ ./view test
contents...
```

답 :

8. [총 8 점/각 2 점] 아래의 문제들은 UNIX 에서 편리하고 강력한 기능을 제공하는 개발 도구(gcc, make)와 관련된 문제들이다.

[1) - 2)] C 의 컴파일 과정은 아래의 그림과 같은 단계를 거친다. gcc 는 이와 같은 컴파일 과정에서 거의 모든 부분을 담당하고 있다. 본래 gcc 가 하는 역할은 preprocessor 와 compiler 의 역할이지만 내부적으로 UNIX 의 assembler 인 as 와 linker 인 ld 를 호출함으로써, 전 과정을 맡고 있다.



아래의 1)과 2)의 물음에 답하시오. test.c 파일은 cs20070887 계정의 홈디렉토리에 존재한다.

1) tset.c 라는 소스파일을 gcc 를 이용하여 컴파일을 하고자 한다. 그런데 컴파일 과정에서 assemble 이전 과정만 수행하여 위 그림에서 보이는 'test.s' 파일을 생성하려고 한다. 사용할 명령어를 아래의 프롬프트 옆에 적으시오.

* cs20070887@cspro ~ \$ _____

2) tset.c 라는 소스파일을 gcc 를 이용하여 컴파일을 하였다. 위 그림에서 보이는 실행 파일 'a.out'을 생성하기 위해 사용할 명령어를 아래의 프롬프트 옆에 적으시오. (명령어를 적을 때 대소문자를 구분하여 적으시오.)

* cs20070887@cspro ~ \$ _____

3) main.c 와 stringM.c 와 output.c 그리고 my.h 로 이루어진 myapp 라는 프로그램을 make 도구를 이용하여 컴파일을 하려고 아래와 같은 makefile 을 작성하였다. 잘못된 부분을 찾아 고치시오.

```
myapp: main.o stringM.o output.o

gcc -o myapp main.o stringM.o output.o

main.o: main.c my.h

gcc main.c

stringM.o: stringM.c my.h

gcc stringM.c

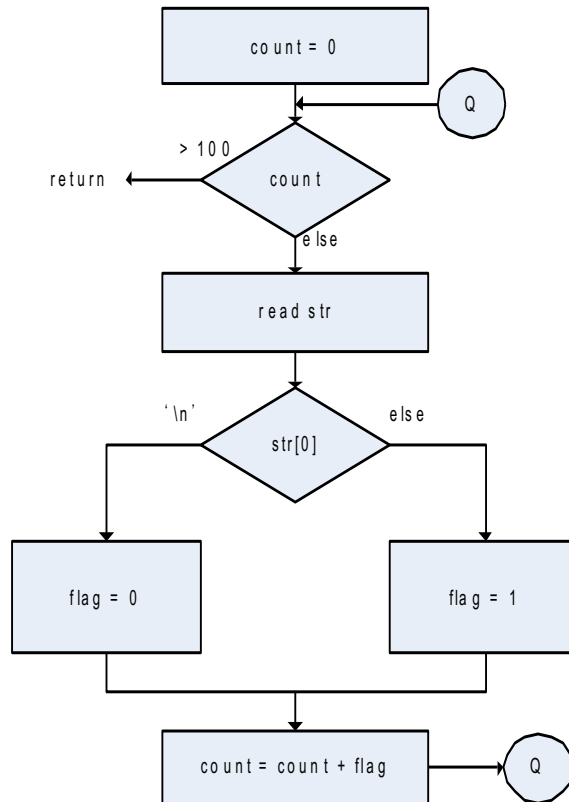
output.o: output.c my.h

gcc output.c
```

4) 다음 중 makefile 을 사용하는 이유와 직접적 관련이 없는 것은?

- ㉠ 다수의 소스 파일로 이루어진 프로그램을 쉽게 컴파일하기 위해서
- ㉢ 다수의 소스 파일로 이루어진 프로그램의 실행 속도를 빠르게 하기 위하여
- ㉡ 다수의 소스 파일로 이루어진 프로그램의 일부만 수정된 경우
수정된 부분에 영향을 받는 파일만 컴파일하여 컴파일 속도를 빠르게 하기
위해서
- ㉣ 한 프로그램에 대하여 버전 별로 다른 컴파일 방법이 존재할 경우
그것들을 저장해두고 필요한 때 골라 쓸 수 있게 하기 위하여

9. [총 7 점] 다음은 test() 라는 함수의 flow chart 이다. 이를 C source code 로 옮겨서 작성하라. (flow chart 아래에 있는 조건을 주의하시오.)



조건:

1. test 라는 함수는 return type 이 void 이고, parameter 를 받지 않는다.
2. count 와 flag 는 integer type 의 local variable 이다.
3. str 는 크기가 30 인 char type 의 array 로서 global variable 이다.
4. flow chart 상의 read str 는 read(str); 이라고 작성한다..

답 :

10. [총 14 점/각 2 점] 다음 “mult.c”파일을 컴파일하여 생성된 실행 프로그램 "app"에 대해 gdb 를 사용하여 디버깅을 하려고 한다.

```
1: #include <stdio.h>
2:
3: int mult(int a, int b){
4:     printf("%d x %d = %dWn", a, b, a*b);
5: }
6:
7: int main(void){
8:     int i, j;
9:
10:    printf("Test ProgramWn");
11:    for(i=2; i<10; i++){
12:        for(j=2; j<10; j++){
13:            mult(i, j);
14:        }
15:        printf("Wn");
16:    }
17: }
```

아래의 과정들은 gdb 를 사용하여 디버깅을 하는 모습을 순차적으로 보여준다. 아래의 설명을 잘 읽고 다음의 명령어들의 빈칸을 완성하시오.

[1) - 2)] 위“mult.c”파일을 컴파일하여 생성된 실행 프로그램 "app"에 대해 gdb 를 사용하여 디버깅을 하려고 한다.

- 1) 컴파일을 할 때 gdb 를 사용하기 위해 사용해야 할 알맞은 명령어를 적으시오.
- 2) gdb 를 실행하기 위하여 사용해야 할 알맞은 명령어를 적으시오.

```
cs20070887@cspro ~ $ gcc -o app 1)_____ mult.c
```

```
cs20070887@cspro ~ $ gdb 2)_____
```

```
GNU gdb 6.4
```

```
Copyright 2005 Free Software Foundation, Inc.
```

```
GDB is free software, covered by the GNU General Public License, and you are  
welcome to change it and/or distribute copies of it under certain conditions.
```

```
Type "show copying" to see the conditions.
```

```
There is absolutely no warranty for GDB. Type "show warranty" for details.
```

```
This GDB was configured as "i686-pc-linux-gnu"...Using host libthread_db library  
"/lib/libthread_db.so.1".
```

```
gdb>
```

[3) - 5)] 디버깅을 시작하기 위하여 프로그램의 내부를 들여다보고 적당한 위치에 breakpoint 를 걸어서 프로그램을 실행 도중에 중단시켜보고자 한다.

- 3) breakpoint 를 설정할 부분을 살펴보기 위하여 아래와 같이 main 함수를 화면에 출력하려 한다. (main 함수가 시작하는 line number 는 현재 모른다.) 사용해야할 알맞은 명령어를 적으시오.(단축키도 가능.)

- 4) breakpoint 를 for 문이 시작하기 전 라인(라인 넘버: 10)에 설정하고자한다.
사용해야할 알맞은 명령어를 적으시오.(단축키도 가능.)
- 5) (4)에서처럼 breakpoint 를 설정하고 프로그램을 실행하면 breakpoint 에서 프로그램이
중단되고 gdb 프롬프트가 나타난다. 프로그램을 실행하기 위한 알맞은 명령어를
적으시오.(단축키도 가능.)

```
gdb> 3)_____
2
3   int mult(int a, int b){
4       printf("%d x %d=%d\n",a,b,a*b);
5   }
6
7   int main(void){
8       int i,j;
9
10      printf("Test Program\n");
11      for(i=2;i<10;i++){

gdb> 4)_____
Breakpoint 1 at 0x80483c1: file mult.c, line 10.

gdb> 5)_____
Starting program: /home/under/app
Breakpoint 1, main () at mult.c:10
10 printf("Test Program\n");
gdb>
```

[6) - 7)] 이제 for 문을 한 라인씩 수행하면서 i 와 j 값의 변화를 살펴보기로 하자.

- 6) i 와 j 변수의 값을 프로그램 수행 중단 때 마다 살펴보기 위하여 프로그램이 실행을
중지 할 때마다 변수 i 와 j 의 값을 출력하고자 한다. 사용해야 할 알맞은 명령어를
적으시오.
- 7) i 와 j 값을 변경하여 mult()함수를 실행하고자 한다. 변수에 값을 할당하는 명령어를
사용해야 한다. 알맞은 명령어를 적으시오.

```
gdb> 6)_____ i

gdb> 6)_____ j
gdb> 한 라인씩 수행하는 명령어 step 또는 next 중 해당하는 명령어
Test Program
11          for(i=2;i<10;i++){
2: j = 0xb8000ca0
1: i = 0x804841c
gdb>
12          for(j=2;j<10;j++){
2: j = 0xb8000ca0
```

```

1: i = 0x2
gdb>
13                mult(i,j);
2: j = 0x2
1: i = 0x2
gdb>
2 x 2=4
12                for(j=2;j<10;j++){
2: j = 0x2
1: i = 0x2

gdb> 7)          i = 4

gdb> 7)          j = 6
gdb> next
13                mult(i,j);
2: j = 0x6
1: i = 0x4
gdb>
4 x 6=24
12                for(j=2;j<10;j++){
2: j = 0x6
1: i = 0x4

```

11. [총 8 점] 실습시간 작성하였던 fmt 프로그램에서 문자열의 끝에 있는 blank 를 제거하는 동작을 하는 함수 remove_end_space()가 있다. (^ : 공백문자)

입력 1:

Programmingneedslogicalthinking. ^ ^ ^ ^ ^ ^ ^ Wn

출력 1:

Programmingneedslogicalthinking.Wn

입력 2:

^ ^ ^ ^ ^ ^ ^ ^ ^ Wn

출력 2:

Wn

이 되도록 아래의 조건을 고려하여 remove_end_space()함수를 작성하라. 단, 아래와 같은 main 함수에서 remove_end_space()함수를 호출한다.

- 1) 실행환경은 UNIX 이다.
- 2) blank 로는 space 만 고려하고 tab 은 고려하지 않음
- 3) 문자열 끝에는 한 개의 개행문자(Wn)만 있음
- 4) return type 은 int 이다. remove_end_space()함수가 return 하는 값은 B_Line flag 를 1 또는 0 으로 set 해준다. (B_Line : 한 줄을 읽고 뒤의 blank 글자를 제거하는 모든 처리 과정을 완료한 후에 그 줄이 빈 줄이라면 B_Line 을 1 로 set 하고 빈 줄이 아니라면 B_Line 을 0 으로 set 한다.)
- 5) main 함수에서 Remove_Blanks(L1) 함수를 호출하기 전에 파일을 열고, L1 과 L2 의 메모리를 할당하고, 파일로부터 문자열 한 줄을 읽어와 L1 에 저장하는 과정을

마친다고 하자. 마찬가지로 Remove_Blanks(L2) 함수를 호출하기 전에 파일로부터 문자열 한 줄을 읽어와 L2 에 저장하는 과정을 마친다고 하자.

- 6) 필요하다면 문자열의 길이를 리턴해주는 **strlen(const char *str)** 함수를 사용하여도 된다.

```
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[])
{
    char *L1,*L2;
    int B_Line=0;
    . . .
    B_Line=Remove_Blanks(L1);
    . . .
    while(1){
        . . .
        Remove_Blanks(L2);
        . . .
    }
    . . .
}
```

답>