

Problem Set 1

Writing, compiling, and debugging programs. Preprocessor macros. C file structure. Variables.

Functions and program statements. Returning from functions.

Problem 1.1

(a) What do curly braces denote in C? Why does it make sense to use curly braces to surround the body of a function?

ANSWER:

Curly braces denotes the end of the scope and it denoted end of block code.

Variables declared are alive only within curly braces called local variable.

(b) Describe the difference between the literal values 7, "7", and '7'.

ANSWER:

7 - interger literals, represents integer value.

"7" - string literals, represents array of characters .

'7'- character literals, represents single character value.

(c) Consider the statement

double ans = 10.0+2.0/3.0-2.0*2.0;

Rewrite this statement, inserting parentheses to ensure that ans = 11.0 upon evaluation of this statement.

ANSWER:

double ans = 10.0 + 2.0 / ((3.0 - 2.0) * 2.0) ;

Problem 1.2

Consider the statement

double ans = 18.0/squared(2+1);

For each of the four versions of the function macro squared() below, write the corresponding value of ans.

1. #define squared(x) x*x - ans 12 - (18.0/2 + 1 * 2 + 1 = 9 + 2 + 1)

2. #define squared(x) (x*x) - ans 3.6 - (18.0/(2 + 1 * 2 + 1)

3. #define squared(x) (x)*(x) - ans `18 - (18.0/(2 + 1) * (2 + 1)

4. #define squared(x) ((x)*(x)) - ans 2 - (18.0/((2 + 1) * (2 + 1))

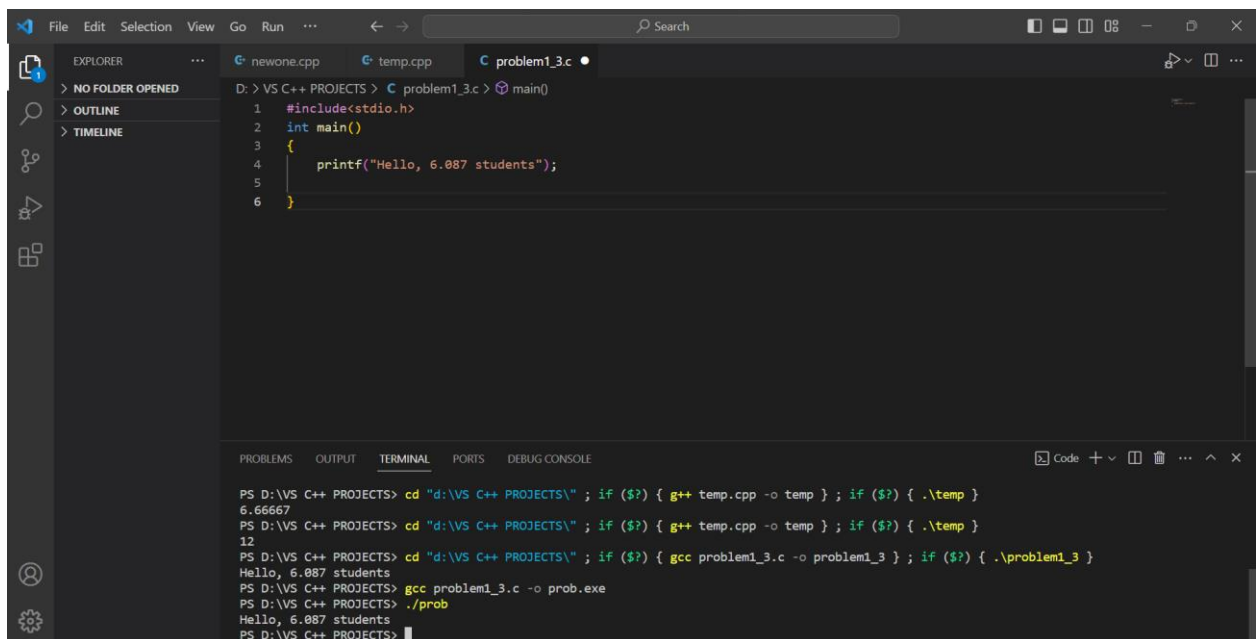
Problem 1.3

Write the “Hello, 6.087 students” program described in lecture in your favorite text editor and

compile and execute it. Turn in a printout or screen shot showing

- the command used to compile your program
- the command used to execute your program (using gdb)
- the output of your program

ANSWER:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'problem1_3.c'. The main editor displays the following C code:

```
D: > VS C++ PROJECTS > C problem1_3.c > main()
1  #include<stdio.h>
2  int main()
3  {
4      printf("Hello, 6.087 students");
5  }
6  }
```

The TERMINAL pane at the bottom shows the following commands and output:

```
PS D:\VS C++ PROJECTS> cd "d:\VS C++ PROJECTS\" ; if ($?) { g++ temp.cpp -o temp } ; if ($?) { .\temp }
6.66667
PS D:\VS C++ PROJECTS> cd "d:\VS C++ PROJECTS\" ; if ($?) { g++ temp.cpp -o temp } ; if ($?) { .\temp }
12
PS D:\VS C++ PROJECTS> cd "d:\VS C++ PROJECTS\" ; if ($?) { gcc problem1_3.c -o problem1_3 } ; if ($?) { .\problem1_3 }
Hello, 6.087 students
PS D:\VS C++ PROJECTS> gcc problem1_3.c -o prob.exe
PS D:\VS C++ PROJECTS> ./prob
Hello, 6.087 students
PS D:\VS C++ PROJECTS>
```

Problem 1.4

The following lines of code, when arranged in the proper sequence, output the simple message "All

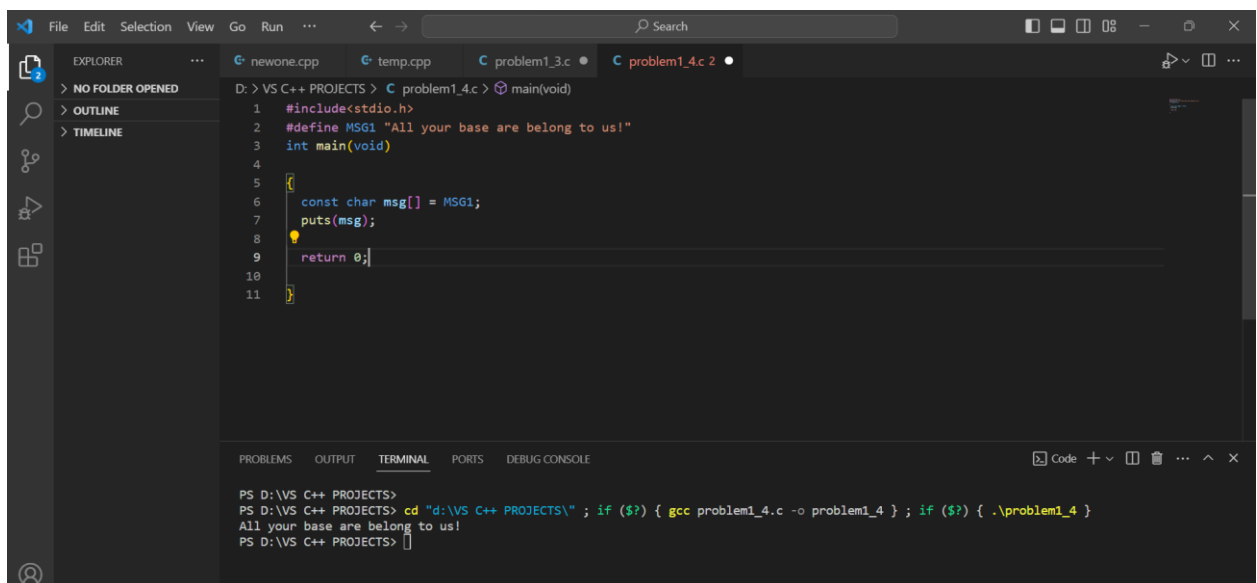
your base are belong to us."

1. return 0;
2. const char msg[] = MSG1;
3. }
4. #define MSG1 "All your base are belong to us!"
5. int main(void) {
6. #include <stdio.h>
7. puts(msg);

Write out the proper arrangement (line numbers are sufficient) of this code.

ANSWER:

6, 4, 5, 2, 7, 1, 3. – correct order



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'VS C++ PROJECTS' with a file 'problem1_4.c' selected. The main editor displays the code for 'problem1_4.c' with the following content:

```
D:\VS C++ PROJECTS> C problem1_4.c > main(void)
1  #include<stdio.h>
2  #define MSG1 "All your base are belong to us!"
3  int main(void)
4
5  {
6      const char msg[] = MSG1;
7      puts(msg);
8
9      return 0;
10
11 }
```

The TERMINAL pane at the bottom shows the command prompt output:

```
PS D:\VS C++ PROJECTS>
PS D:\VS C++ PROJECTS> cd "d:\VS C++ PROJECTS\" ; if ($?) { gcc problem1_4.c -o problem1_4 } ; if ($?) { .\problem1_4 }
All your base are belong to us!
PS D:\VS C++ PROJECTS>
```

Problem 1.5

For each of the following statements, explain why it is not correct, and fix it.

(a) `#include <stdio.h>;`

Preprocessor macros like `#include` should not be terminated with semicolons.
`#include<stdio.h>`

(b) `int function(void arg1)`
`{`
`return arg1-1;`
`}`

Void data type does not hold any data type. Void cannot be used as function parameter.
`int function (int arg1)`
`{`
`return arg1 -1;`
`}`

(c) `#define MESSAGE = "Happy new year!"`
`puts(MESSAGE);`

`#define` statement does not include an assignment operator
`#define MESSAGE "Happy new year!"`
`puts (MESSAGE);`