Name: TIEN DUNG LE (1002024297)

**Problem 1 – 12 pts.**

What does the following code print?

#include <stdio.h>

int main(void)

{

int arr1[] = {-1, 24, 3};

int arr2[] = {12, -10, 7};

int arr3[3];

if (arr1[0] > 0)

{

if (arr2[0] > arr1[0])

{

arr3[0] = arr2[0] - arr1[0];

arr3[2] = arr1[1] + arr2[1];

}

else

{

arr3[0] = arr2[0] - arr1[0];

arr3[2] = arr1[1] + arr2[1];

}

arr3[2] = arr1[2] \* arr2[2];

}

else

{

if (arr1[0] < arr2[0])

arr3[0] = arr1[0];

else

arr3[0] = arr2[0];

if (arr1[1] < arr2[1])

arr3[1] = arr1[1];

else

arr3[1] = arr2[1];

if (arr1[2] < arr2[2])

arr3[2] = arr1[2];

else

arr3[2] = arr2[2];

}

printf("%d %d %d", arr3[0], arr3[1], arr3[2]);

return 0;

}

OUTPUT:

-1 -10 3

**Problem 2 – 10 pts**

What does the following code print?

#include <stdio.h>

int main()

{

int x = 8;

int y = 5;

switch (x-y)

{

case 1:

x += 4;

break;

case 3:

y -= 2;

case 5:

x--;

break;

default:

y++;

}

printf("x: %d\n", x);

printf("y: %d\n", y);

return 0;

}

OUTPUT:

X: 7

Y: 3

**Problem 3 – 12 pts.**

What does the following code print?

#include <stdio.h>

#include <string.h>

int main()

{

char s1[] = "Hello";

char s2[] = "hello";

char s3[] = "1310 - C";

if (!strcmp(s1, s2) == 0)

{

s1[0] = s2[0];

}

if (strcmp(s2, s3) > 0)

{

s3[4] = '\0';

}

else

{

s3[2] = '2';

}

printf("%s\n%s\n%s\n", s1, s2, s3);

return 0;

}

OUTPUT:

hello

hello

1310

**Problem 4 – 18 pts**

In a file called problem4.c, write a program that:

* Asks the user to enter an integer. It is okay for you program to crash when the user does not enter a valid integer.
* If the integer is less than 0, it prints: The number is negative.
* If the integer is divisible by 2 and by 3, it prints: The number is even and divisible by 3.
* If the integer is divisible by 2 but not by 3, it prints: The number is even and not divisible by 3.
* If the integer is divisible by 3 but not by 2, it prints: The number is odd and divisible by 3.
* If the integer is divisible by neither 2 nor 3, it prints: The number is odd and not divisible by 3.

For example: if the user enters -5, your program output should look EXACTLY like this:

Please enter an integer: -5

The number is negative.

As another example: if the user enters 123, your program output should look EXACTLY like this:

Please enter an integer: 123

The number is odd and divisible by 3.

As another example: if the user enters 40, your program output should look EXACTLY like this:

Please enter an integer: 40

The number is even and not divisible by 3.

As another example: if the user enters 41, your program output should look EXACTLY like this:

Please enter an integer: 41

The number is odd and not divisible by 3.

As another example: if the user enters 42, your program output should look EXACTLY like this:

Please enter an integer: 42

The number is even and divisible by 3.

**Problem 5 – 18 pts**

In a file called problem5.c, write a program that:

* Asks the user to enter a word.
* If the word starts with a vowel, the program prints that it starts with a vowel.
* If the word starts with a consonant, the program prints that it starts with a consonant.
* If the work starts with neither a vowel nor a consonant (e.g. it starts with a number, or punctuation), the program prints that it starts with neither a vowel nor a consonant.

For example: if the user enters "543", your program output should look EXACTLY like this:

Please enter a word: 543

543 starts with neither a vowel nor a consonant.

As another example: if the user enters "Airplane", your program output should look EXACTLY like this:

Please enter a word: Airplane

Airplane starts with a vowel.

As another example: if the user enters "cat", your program output should look EXACTLY like this:

Please enter a word: cat

cat starts with a consonant.

**Problem 6 – 30 pts.**

This is the most challenging one. For this program, you must use the switch statement to check the menu choice, not an if statement. You are welcome to use if statements as part of a switch case, but not for the menu choice itself. In a file called problem6.c, write a program that:

* Creates an array of four numbers.
* Asks the user for these numbers.
* Prints the array of numbers. See the example output below for the format.
* Prints a menu for the user. See the example output for the menu.
* Asks the user for their menu choice.
* If the user chooses 1:
  + Ask the user which array position they want to change.
  + If the array position is valid, get the new value from the user and update the array.
  + If the array position is not valid for an array of four elements, print an error message and stop the switch statement.
* If the user chooses 2:
  + Ask the user which array position they want to print.
  + If the array position is valid, print the number stored at that position. See the example output for the format.
  + If the array position is not valid for an array of four elements, print an error message and stop the switch statement.
* If the user chooses 3:
  + Ask the user which positions they want to swap. The user should enter the first position, followed by a space, followed by the second position (e.g. 2 0).
  + If the positions are valid, swap the values stored at those positions.
  + If either position is invalid, print an error message and exit the switch.
* If the user chooses an invalid menu selection (i.e. not 1, 2, or 3), the program should print that it is an invalid menu choice.
* After performing the menu operation or exiting the switch due to an invalid position or menu choice, print the array values again.

Note that if you try to declare a variable as part of a switch case, the compiler will complain. One way around this is to declare any variables you need before the switch statement. For example:

int position;

switch (…)

{

case 1:

// you can use position here.

case 2:

// position is available here too.

}

Another way around it is to put the statements for each case in a block (put curly braces around them), like this:

switch (…)

{

case 1:

{

int position;

// Use position in this case.

}

case 2:

{

int position;

// Use position in this case.

}

}

Here are some example runs from my implementation, so you can get the output formatting correct. I am not using any field width or precision with my format specifiers.

**Example output 1 (invalid menu choice):**

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

Enter the fourth number: 4

Array: [1.000000, 2.000000, 3.000000, 4.000000]

Menu

1: Change a number

2: Print a number

3: Swap array positions

Enter your choice: 5

Invalid menu choice.

Array: [1.000000, 2.000000, 3.000000, 4.000000]

**Example output 2 (invalid array position):**

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

Enter the fourth number: 4

Array: [1.000000, 2.000000, 3.000000, 4.000000]

Menu

1: Change a number

2: Print a number

3: Swap array positions

Enter your choice: 1

Which position do you want to change (0, 1, 2, 3)? 4

Invalid position.

Array: [1.000000, 2.000000, 3.000000, 4.000000]

**Example output 3 (menu option 1):**

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

Enter the fourth number: 4

Array: [1.000000, 2.000000, 3.000000, 4.000000]

Menu

1: Change a number

2: Print a number

3: Swap array positions

Enter your choice: 1

Which position do you want to change (0, 1, 2, 3)? 0

To what number do you want to change position 0? 5

Array: [5.000000, 2.000000, 3.000000, 4.000000]

**Example output 4 (menu option 2):**

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

Enter the fourth number: 4

Array: [1.000000, 2.000000, 3.000000, 4.000000]

Menu

1: Change a number

2: Print a number

3: Swap array positions

Enter your choice: 2

Which position would you like to print (0, 1, 2, 3)? 3

Value at index 3: 4.000000

Array: [1.000000, 2.000000, 3.000000, 4.000000]

**Example output 5 (menu option 3):**

Enter the first number: 1

Enter the second number: 2

Enter the third number: 3

Enter the fourth number: 4

Array: [1.000000, 2.000000, 3.000000, 4.000000]

Menu

1: Change a number

2: Print a number

3: Swap array positions

Enter your choice: 3

List the positions you want to swap separated by a space (e.g. 0 3): 2 3

Array: [1.000000, 2.000000, 4.000000, 3.000000]

It might help to break this up into manageable chunks. Implement one chunk at a time and test it. For example, the first task is to generate the array, so write the code to get the numbers from the user and print the array. Test it. Then write the code to generate the menu. Make sure it prints correctly. Then implement menu choice 1 and test it, and so on. Build it piece by piece, testing along the way. You can do this!

**Submission:**

Place this Word document and the three program source files (problem4.c, problem5.c, and problem6.c) in a folder and zip the folder. Submit the zip file in Canvas. Make sure that the zip file contains your source code and Word document before submitting.