Popquiz! 12 Easy Pieces

The last popquiz 3 weeks ago was probably too difficult. This contains more problems, but simpler ones. Solving these will also get you very close to solving the current assignment and bonus assignments.

1. Write function documentation for the following function, which gets input from the keyboard, puts it into a one-dimensional array, and prints the array.

```
// Name: input
// Purpose: Store and print integer array values
// Returns: Nothing (void)
// Params: Integer array [], integer size of array
void input(int [], int);
```

2. Write a function `sum` that takes an integer array `foo` of length `n` sums up its elements, and returns the sum of the elements.

```
int sum(int foo[], int n)
{
  int sum=0;
  for (int i=0;i<n;i++)
    sum+=foo[i];
  return sum;
}

int main(void)
{
  int array[5]={1,1,1,1,1};
  printf("Sum = %d\n",sum(array,5));
  return 0;
}
```

```
Sum = 5
```

3. Write a function `print` that prints an array `bar` of length `k` using a `for` loop.

```
void print(int bar[], int k)
{
  int j;
  for(j=0;j<k;j++)
    printf("%d \n", bar[j]);
}

int main(void)
{
  int array[5]={1,1,1,1,1};
  print(array,5);
  return 0;
}
```

```
1
1
1
```

```
1
1
```

4. Write a statement that prints a long integer variable `i`, a floating-point variable `x` to precision 4, and a Boolean variable `b`.

```c
#include <stdbool.h>
int i = 100;
float x = 3.141592;
bool b = true;

printf("Integer i = %d, float x = %g, bool b = %d\n", i, x, b);
```

```
Integer i = 100, float x = 3.14159, bool b = 1
```

5. Declare a floating-point constant `PI` with the value `3.141592`.

```c
#define PI 3.141592  // preprocessor directive
const float pi2 =  3.141592;  // const definition
```

6. Comment each line of this code:

```c
#include <stdio.h> // include Input/Output library

int main(void)  // main function (no arguments, returns integer

{  // begin of main program/function

  int time;  // declare integer variable

  printf("Enter time in seconds:\n"); // Ask for user input

  scanf("%d",&time); // get user input as address-of variable

  float answer = (32 * time * time) / 2; // compute float result

  printf("\nThe distance is %g feet.\n", answer); // print result

  return 0; // return 0 if program runs successfully

} // end of main program / function
```

```
Enter time in seconds:

The distance is 2.68435e+08 feet.
```

7. Write a nested `for` loop to iterate over a 2 x 2 integer array `baz`. Inside the `for` loop, print an array element.

```c
int baz[2][2]={1,2,3,4};
for (int i=0;i<2;i++) {
  for (int j=0;j<2;j++) {
    printf("%d ", baz[i][j]);
  }
```

```
  puts("");
 }
```

```
1 2
3 4
```

8. Change the following function so that the parameter `signal` can be passed by reference from the calling function:

   Pass-by-value:

```
void morse ( char  signal )
{
  printf("Your morse signal was '%c'\n",  signal );
}
int main(void)
{
  char c = '-';
  morse(c);
  return 0;
}
```

```
Your morse signal was '-'
```

   Pass-by-reference:

```
void morse ( char* signal )
{
  printf("Your morse signal was '%c'\n", *signal );
}
int main(void)
{
  char c = '-';
  morse(&c);
  return 0;
}
```

```
Your morse signal was '-'
```

1. Write a statement that prints `i`, `p` and `**p` below, and write after the `//` comments what `i`, `p`, and `pp` each are.

```
int i  = 100;
int* p = &i;
int** pp = &p;
// print i, p, **pp
printf("i = %d, p = %d, pp = %d\n",i,p,**pp);
```

```
i = 100, p = -1173115052, pp = 100
```

2. For given integers `a` and `b`, complete the selection statement below:

```
int a = 10, b = 100;

if (a == b) {
        printf("%d and %d are the same!\n",a,b);
  } else if (a > b) {
        printf("%d is bigger than %d!\n",a,b);
  } else {
        printf("%d is smaller than %d!\n",a,b);
  }
```

```
10 is smaller than 100!
```

3. Write a `while` loop that prints your name 10 times before exiting:

```
int i = 0;
while (i < 10) {
  printf("Marcus ");
  i++;
 }
```

```
Marcus Marcus Marcus Marcus Marcus Marcus Marcus Marcus Marcus Marcus
```

4. You've tangled a source code file as `main.c`. Write the code necessary to compile it on the command-line so that the executable is called `a.out`

```
gcc main.c -o a.out
gcc main.c
```

What if you didn't do well on this popquiz? There are several possibilities:

1. You don't do well under pressure. To remove this obstacle, do the quiz all over (without peeking at the solutions): tinyurl.com/functions-popquiz.
2. You just don't have the basics to solve most of the questions. Then you must work through the solutions and make sure you understand it, and work through the first 15 chapters of w3schools.com for C (free and online).
3. You (think you) know the basics, and you understood the review of the quiz, but you're just too darn slow. Then you need to solve more small problems, starting with the bonus problems in the course, and you can always ask me for more small problems.

Created: 2024-10-07 Mon 23:19