# Take Notes

IPC144 Practice Exam 20151  
  
Write all answers on a separate set of papers.

Short Answer Questions (10 Marks)

**Q1)** Explain the term “control structure”, what they are used for and list 3 examples: **[3 Marks]**

**Answer:** The term “control structure” defines as the statements which we control the behavior of our program and to control the flow of execution in a program. We used “control structure” to alter the program execution and alter the direction of the program in one or more paths available.

Examples of control structures are:

* Sequence
* Selection
* Iteration

**Q2)** Explain the difference between a string and a character array: **[2 Marks]**

**Answer:** String is a collections of character with special property called null terminator (“\0”) as last character, where as character array are set of element which use space as escape character. String will have static storage duration, but character array will not any storage unless user defined.   
  
**Q3)** How is code modularity achieved in the C language, list some examples: [**3 Marks]**

**Answer:** Modularity is a software or program design technique that emphasizes separating the functionality of a program into independent, interchangeable modules into a complete solution.

Example:

Cohesion – A highly cohesive module performs a single task and only that task.

Coupling – A inter-dependability among other modules   
  
**Q4)** What is the danger of returning the address of a local variable from within a function? **[2 Marks]**

**Answer:**

Programming Questions (20 Marks)  
 **Q5)** Write a function which will receive a string and inner cap its contents. The function should receive a constant pointer to a null terminated character string [source], a pointer to a character array [destination], and return an integer. The function should read the source array and copy the characters into the destination array, capitalizing the first letter of each word in the string (separated by spaces). The destination must be correctly null terminated after the operation is complete. This function returns the number of words encountered in the source string. **[5 Marks]**  
  
Example:   
“hello world, do you remember me?”   
Should be be converted to   
“Hello World, Do You Remember Me?”

**Q6)** Write an inventory management program that keeps an up to date listing of the items in inventory in a file. When the program starts up, it should read an inventory file called “inventory.txt” located in the same directory as the program. This file contains the listing of inventory items. Each item contains a name and day to expire field separated by a colon ( : ) and each item is separated by a newline. The item name will be no more than 20 characters long.

The program should read the file and store each item in memory. The program should not store more than 100 items. Once loaded, the program should display a menu to the user and prompt them for a selection. The menu should offer to: Add item, List items, Decrement expiry days, or Quit. The program should loop until the user has indicated they wish to quit.

To add an item, prompt the user for a new item, requiring the user to enter name and days to expire as a string and integer separated by a colon (:). Ensure the string accepted is 20 characters or shorter and expiry is above 0. If the name entered is too long, ignore characters beyond the limit.  
  
To list items, display the item information separated by a space, each item being on a new line.

To decrement expiry days, reduce the number of days to expire by one on all items. If any item expiry reaches zero, remove it from the list. A removed item should not appear when the user requests an item listing and it should not be saved to file when the user quits.

When the user chooses to quit the program, the program should open the inventory.txt file and write out the list of items currently stored in memory in the same format as it was loaded.  
  
Validate input by displaying an error and looping if the user inputs an invalid value. **[15 Marks]**

|  |  |
| --- | --- |
| Program Output | Inventory.txt |
| [1 - Add Item, 2 - List Items, 3 - Decrement expiry, 4 - Quit] : 0 Invalid try again: 1 Enter item info: Carrot Cake:-1 Invalid try again: Carrot Cake:3 [1 - Add Item, 2 - List Items, 3 - Decrement expiry, 4 - Quit]: 2 string beans 9  can of worms 1  bananas 16 Carrot Cake 3 [1 - Add Item, 2 - List Items, 3 - Decrement expiry, 4 - Quit]: 3 Decremented… [1 - Add Item, 2 - List Items, 3 - Decrement expiry, 4 - Quit]: 4 File updated! | |  | | --- | | At Startup | | string beans:9  can of worms:1  bananas:16 | | After Run | | string beans:8  bananas:15 Carrot Cake:2 | |

**Q7)** Write out the exact output for this program **[10 Marks]**

int hertom(int\* remorse, int primer)

  {

  (\*remorse)++;

  return \*remorse > primer;

  }

void anton(int\* administer, int chalice)

  {

  \*administer %= chalice;

  }

void centration(int\* refuse, int burst)

  {

  int i;

  for(i = 0; i < burst - 1; i++)

     {

     if(hertom(&refuse[i], refuse[i + 1]))

        {

        anton(&refuse[i], refuse[i + 1]);

        printf("Whispers\n");

        }

     else

        {

        printf("Contaminants\n");

        }

     }

  }

int main()

  {

  int ar[] = {3, 9, 4, 15, 21, 9, 12, 98, 43, 4, 26, 71};

  int\* hop = ar;

  int i = 0;

  for(i = 0; i < 12; i++)

     printf("%d - ", hop[i]);

  puts("\n");

  centration(ar, 10);

  puts("\n");

  for(i = 0; i < 12; i++)

     printf("%d - ", hop[i]);

  }

## To learn more and get OneNote, visit [www.onenote.com](http://go.microsoft.com/fwlink/?LinkID=523891).