

Soft2129 Sample Exam Questions

These questions are from previous exams. They are grouped according to unit postconditions.

1. Question on basic C.

- Write a C program that copies only the even numbered lines from input to output. So, for a 7 line file, it copies the second, fourth, sixth lines.
- other tasks as in the Week 4 prac quiz and from the prac in weeks 1--4.

2. Question on basic pointers.

Consider the following code segment:

```
#include <stdio.h>
```

```
/* IncN:
```

```
** Preconditions: num and inc are defined
```

```
** Postconditions: num has been incremented by the amount 'inc'
```

```
*/
```

```
void IncN(int num, int inc)
```

```
{
```

```
    num = num + inc;
```

```
}
```

```
int main()
```

```
{
```

```
    int val = 18;
```

```
    printf("My original number is %d\n", val);
```

```
    IncN(val, 7);
```

```
    printf("My new number is %d\n", val);
```

The author of the code was surprised that the second printf did not give the result 25, as they had expected. In the space below, draw pictures of the memory as it would be at the end of the assignment statement in IncN. Label the parts of the diagram with the lvalue and rvalue of the integer variable, val.

Now draw pictures to show the memory as it will be after the second print of main. Add notes to explain why this code fails to print 25.

Now write changes to the supplied code above to make it work as intended.

3. *Question on pointers with arrays.*

Consider the following code:

```
#include <stdio.h>

int nums[ ] = {18, 2, 11726, -1, 29, 0, 17, 11};
int *p;

int main(void)
{
    int j;
    p = &nums[0];

    for (j = 0; j < 4; j++)
    {
        printf("%d : %d, %d \n", j, nums[j], *(p + j));
    }
    return 0;
}
```

What does this print?

Describe what would be printed if `*(p + j)` were replaced by `*p++` .

Draw a picture of the memory of the original code as it is at the last printf. Show which data is on the stack, heap, global/static/extern/constant areas of memory.

4. *Questions on dynamic allocation*

You are given the following definition for an element of a list.

```
typedef struct Node {
    char *name;
    struct Node *next;
} Node;
```

```
Node *start;
```

Assume that a list of `Node` has been constructed by dynamically allocating each list element using `malloc`, and `start` points to it. The `next` field points to the next element in the list for all but the last element, where it is `NULL`.

a) Write a C function `Node *DeCap(Node *start)` which removes the entries in the list

Soft2129 Sample Exam Questions

where the names start with a lower case letter, where **start** points to the beginning of the list. On completion, that list should contain all those elements from the original list except those that have a lower case at the beginning of the name. It should free all the memory associated with the parts of the list that have been deleted.

b)

Write a function `Node *reverse(Node *reverseMe)` which returns a new list that contains exactly the same elements as the one pointed to by `reverseMe` but in reverse order and it deletes the original list.

5. Questions on basic unix

- Jamie runs his C code with the command `a.out` Albert finds this does not work but `./a.out` does work. What would you tell Albert about the likely cause of the difference?

How could Albert change things so that he can run this program using just `a.out` as Jamie can?

- You see a John compiling their code using

```
gcc code.c
```

even though the class notes recommend

```
gcc -Wall -W -ansi -pedantic -g code.c -o code
```

What would you tell John about the benefits of the recommended line?

- Amy is new to unix and asks you to explain some file system concepts she has heard about. In the space below, draw a depiction of the unix file system and label it to explain the following concepts to Amy:
 - root
 - home directory
 - /
 - /usr/pub/ascii
 - current directory
 - .

Soft2129 Sample Exam Questions

- ..
- full path name
- relative path name
- \$HOME
- search path for shell

6. Question that integrates elements

Given the following definition:

```
typedef struct {
    char *name;
    int Age;
    float Weight;
} Person_info;
```

Complete the function with the header:

```
/* Ask the user their name, age and weight
** returns a pointer to the Person_info structure with this information
*/
```

```
Person_info * ask_user()
{
}
```

Write a main function and any other functions you think necessary to read name/age/weight information from input using "ask_user", placing the details into a linked list of "Person_info". Stop when an empty name is entered.

Then construct an array of pointers to the list elements and sort the array according to ascending order of weight. Then print the elements of the sorted array, one per line, name/age/weight.