

CSCI-UA.0201
Computer System Organization
Homework Assignment 1
(Maximum grade: 40)

1. [1 point] What will happen if, in a C program, you assign a value to an array element whose index exceeds the size of array (e.g. $A[15] = 8$ while A is an array of 14 elements only)? You must specify what will be the compiler reaction, and, if the compilation is successful, what will happen during execution.

2. [1 point] In C, if you pass an array as an argument to a function, what actually gets passed?

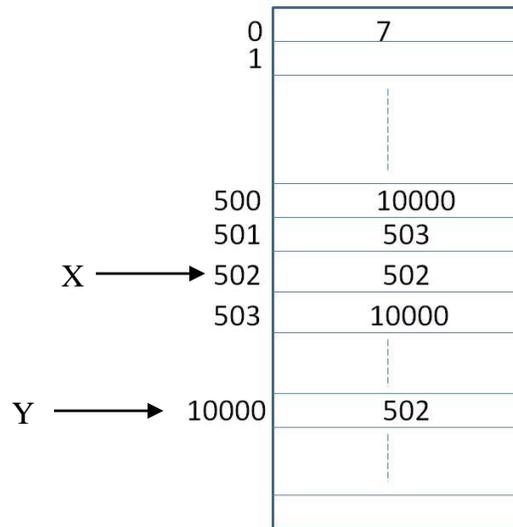
3. [6 points] What will be the output of the following program? Explain how you reached your solution.

```
#include<stdio.h>

int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int x, y, z;
    x = ++a[1];
    y = a[1]++;
    z = a[x++];
    printf("%d, %d, %d", x, y, z);
    return 0;
}
```

4. [1 point] In C, you have to free the memory you allocated. In language like Java, the garbage collector is doing this for you. State one advantage for each strategy.

5. [5 points] Assume we have the following memory map:



and we also have the following variable declarations:

```
char x = 502; /* x will be stored in address 502 */  
char * y;
```

Then we execute: `y = &x;` /* Let's assume y is stored in address 10000 */

- What is the value of: `&x` ?
- What is the value of: `y` ?
- What is the value of `*y` ?
- What is the value of `&y` ?
- What is the value of `*(y)` ?

6. [2 points] Is there anything wrong with the following declaration? If so, what is it?

```
struct emp  
{  
    int ecode;  
    struct emp *e;  
};
```

7. [2 points] Is there anything wrong with the following code? If so, what is it?

```
struct emp
{
    int ecode;
    struct emp e;
};
```

8. [2 points] Assume we are executing the following program on a 64-bit machine:

a) What will be the output?

b) If we change p's declaration to be: `double * p;` and use the type casting with `malloc` to be `(double *)`, what will be the output?

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

int main()
{
    int *p;
    p = (int *)malloc(20);
    printf("%d\n", sizeof(p));
    free(p);
    return 0;
}
```

9. [4 points] State 2 advantages for using pointers.

10. We said in class that returning the address of a local variable is wrong.

a) [1 point] Why is that?

b) [1 point for 1st part, 4 points to justification] In the following code x is a local variable. Is there anything wrong with that code? Justify

```
int * memory()
{
    int * x;

    x = (int *)malloc(sizeof(int)* 1000);
    return x;
}
```

11. We have seen that with n bits we have 2^n patterns.

- a) [2 points] Why is the largest unsigned number that can be presented with n bits (2^n-1) and not 2^n ?
- b) [2 points] Why is the largest positive number that can be presented in signed numbers $(+2^{n-1}-1)$ and not $(+2^{n-1})$?
- c) [1 point] For signed numbers of n -bits, do we have more negative numbers or more positive numbers?

12. Suppose we have this binary number: 10101010

- a) [1 point] Interpret this number as unsigned number and write the decimal equivalent.
- b) [1 point] Interpret this number as signed number and write the decimal equivalent.
- c) [1 point] Write the number in hexadecimal.
- d) [2 points] Does the hexadecimal presentation depend on whether the number is signed or unsigned? Why?