

1. The following code snippet was written to copy one C string's (str1) contents to another (str2), but it doesn't work. Write the correct code.

```
char str1[100] = "Hello, world!";
char str2[100] = "";
str2 = str1;
```

2. You are given the following code. Make a picture showing the structure of the dynamically allocated variables. Also fill in the gaps in the code as instructed in the comments and give the output of the program. (5 + 5 + 5 points)

```
int main(void)
{
    int ss[] = {3,5,4,1,2};

    int ** darray = new int*[5];
    darray[0] = new int(7);

    int i, j;
    for(i=1; i<5; i++)
        darray[i] = new int[ss[i]];

    for(i=1; i<5; i++)
        for(j=0; j<ss[i]; j++)
            darray[i][j] = i+j;

    cout << *darray[0] << endl;
    for(i=1; i<5; i++)
    {
        for(j=0; j<ss[i]; j++)
            cout << darray[i][j] << " ";
        cout << endl;
    }

    /* deallocate all the dynamically allocated variables */

    return 0; }
```

Picture showing the structure of the dynamically allocated variables:

Output of program:

3. Design a class that represents experimental readings, called **Readings**. It contains an array (of doubles) of some size that will store the values obtained during some experiment. When an object of this class is initialized, the programmer will specify how many elements the array should contain. Your class should create the array at that point.

The object cannot be initialized without this value. Also, we do not want to give direct access to the array at any point.

The class should contain the following functions, in addition to the constructor(s):

- `bool addReading(double reading)`  
This adds a value to the array. If the array is already full, print a message and return false. Otherwise, return true.
- `double getMean()`  
Returns the average of the values *currently* in the array

Your class should have another function, with the signature `double meanSquaredError(Readings r)`, which computes the mean squared error between the arrays of the current object and the object `r`. The mean squared error is computed in the following way: first find the differences between two array elements at each index. Then find the squares of those quantities. Then sum them up. Finally, divide the sum by the number of elements in the array. Mathematically, the formula looks like this:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\text{array1}[i] - \text{array2}[i])^2$$

You do not have to write the getters and setters for your class, but you may assume they exist. You should, however, demonstrate separation of interface and implementation.

4. Describe in a sentence or two what this function does:

```
void magic( char * string1 , const char * string2 , int string2size )
{
    *( string1 + string1size ) = '\0' ;
    string1 = string1 + string1size - 1;
    for ( ; *string2 != '\0' ; string1--, string2++ )
        *string1 = *string2 ;
}
```

5. Write a function that takes a c string (array) as a paramter and prints everything in the string after the first occurrence of the letter “x”. If “x” is not in the string, it prints nothing.
6. A student is writing a program that deals with records of books. She has defined two structs and now you have to write some parts, too. Assume that the title of any book is no longer than 99 characters, and a person’s first or last name is no longer than 49 characters. (7 + 4 + 7 + 7 points)

```
struct PersonName {
    char first[50];
    char last[50];
};

struct Book {
    char title[100];
    PersonName author;
    int numOfPages;
};
```

Write the function `makeBook`, which creates a `Book` element on the heap, fills it with the relevant values (which are passed as parameters) and returns the pointer.

*Pointers* to these dynamic `Book` objects will need to be stored in an array. Write a function `initArray(int len)` which allocates a dynamic array, of length `len`, of `Book pointers`, and returns a pointer to the array. State the signature clearly.

Next, a function to insert book pointers into the array needs to be written. The function `addBook()` takes four parameters: A pointer to the dynamic book object (`book`), a pointer to the array (`array`), the position at which `book` will be stored in the array (`pos`), and the maximum length of the array (`maxLen`). If `pos` is invalid, the function returns false, else it adds the pointer in the array and returns true. The function also increments `pos`, and the change is reflected in the original value that was sent as the parameter `pos`. Write this function and its signature.

Finally, we need a function to print the title, author's name and number of pages of each book in the array. Write it down. Your function can have any parameters you want, but make sure you state the signature clearly.