Preprocessor directive Lab

```
//Macro to find the biggest of two numbers and implement array sorting

#include <stdio.h>

#define MAX(x, y) ((x) > (y) ? (x) : (y))

void bubbleSort(int arr[], int size) {

    int i, j, temp;

    for (i = 0; i < size - 1; i++) {

        for (j = 0; j < size - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

int main() {

    int num1 = 10, num2 = 20;

    int sortedArray[] = {5, 1, 4, 2, 8};

    int size = sizeof(sortedArray) / sizeof(sortedArray[0]);

    int i;

    printf("Biggest of %d and %d is %d\n", num1, num2, MAX(num1, num2));

    printf("Unsorted array: ");

    for(i=0; i<size; i++){

        printf("%d ", sortedArray[i]);

    }
    printf("\n");
      bubbleSort(sortedArray, size);
    printf("Sorted array: ");
    for (i = 0; i < size; i++) {
        printf("%d ", sortedArray[i]);
    }
    printf("\n");
    return 0;
}
//Macro to find the biggest of four numbers using the biggest of two macro
#include <stdio.h>
#define MAX(x, y) ((x) > (y) ? (x) : (y))
#define MAX4(a, b, c, d) MAX(MAX(a, b), MAX(c, d))
int main() {
    int a = 10, b = 20, c = 30, d = 40;
    printf("Biggest of %d, %d, %d, and %d is %d\n", a, b, c, d, MAX4(a, b, c, d));
    return 0;
}
//3. Program to print if a character is alphanumeric or a special character using macro conditions
#include <stdio.h>
#define IS_ALPHANUMERIC(ch) (((ch) >= 'a' && (ch) <= 'z') || ((ch) >= 'A' && (ch) <= 'Z') || ((ch) >= '0' && (ch) <= '9'))
int main() {
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);
    if (IS_ALPHANUMERIC(ch)) {
        printf("%c is an alphanumeric character.\n", ch);
    } else {
        printf("%c is a special character.\n", ch);
    }
    return 0;
}
//Macro to print the elements of an array
#define PRINT_ARRAY(arr, size) \
    do { \
        int i; \
        printf("Array elements: "); \
        for (i = 0; i < (size); i++) { \
            printf("%d ", (arr)[i]); \
        } \
        printf("\n"); \
    } while (0)
int main() {
    int intArray[] = {1, 2, 3, 4, 5};
    float floatArray[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    char charArray[] = {'a', 'b', 'c', 'd', 'e'};
    int intSize = sizeof(intArray) / sizeof(intArray[0]);
    int floatSize = sizeof(floatArray) / sizeof(floatArray[0]);
    int charSize = sizeof(charArray) / sizeof(charArray[0]);
    PRINT_ARRAY(intArray, intSize);
    PRINT_ARRAY(floatArray, floatSize);  // Note:  Will print as integers.
    PRINT_ARRAY(charArray, charSize);    // Note:  Will print as integers.

    return 0;

}

// Generic function to print different types of arrays

#include <stdio.h>

Void printArray(void *arr, int size, size_t elementSize, const char *formatSpecifier) {

    Int i;

    Printf(“Array elements: “);

    For (i = 0; i < size; i++) {

        // Use a pointer to access the element at index i

        Void *elementPtr = (char *)arr + i * elementSize;

        // Print the element using the provided format specifier


        Printf(formatSpecifier, *( (typeof(int)*)elementPtr)); //added a typecast

        Printf(“ “);

    }

    Printf(“\n”);

}

Int main() {

    Int intArray[] = {1, 2, 3, 4, 5};

    Float floatArray[] = {1.1, 2.2, 3.3, 4.4, 5.5};

    Char charArray[] = {‘a’, ‘b’, ‘c’, ‘d’, ‘e’};

    Int intSize = sizeof(intArray) / sizeof(intArray[0]);

    Int floatSize = sizeof(floatArray) / sizeof(floatArray[0]);

    Int charSize = sizeof(charArray) / sizeof(charArray[0]);

    printArray(intArray, intSize, sizeof(int), “%d”);
    printArray(floatArray, floatSize, sizeof(float), “%f”);

    printArray(charArray, charSize, sizeof(char), “%c”);

    return 0;

}

//Macro to generate swapping functions for different data types

#include <stdio.h>

#define SWAP(type, a, b) \

    Do { \

        Type temp = (a); \

        (a)  = (b); \
        (b)  = temp; \

    } while (0)

Int main() {

    Int int1 = 10, int2 = 20;

    Float float1 = 1.1, float2 = 2.2;

    Double double1 = 10.10, double2 = 20.20;

    Char char1 = ‘a’, char2 = ‘b’;

    Printf(“Before swapping:\n”);

    Printf(“int1 = %d, int2 = %d\n”, int1, int2);

    Printf(“float1 = %f, float2 = %f\n”, float1, float2);

    Printf(“double1 = %lf, double2 = %lf\n”, double1, double2);

    Printf(“char1 = %c, char2 = %c\n”, char1, char2);

    SWAP(int, int1, int2);

    SWAP(float, float1, float2);

    SWAP(double, double1, double2);

    SWAP(char, char1, char2);
    Printf("\nAfter swapping:\n");

    Printf("int1 = %d, int2 = %d\n", int1, int2);

    Printf("float1 = %f, float2 = %f\n", float1, float2);

    Printf("double1 = %lf, double2 = %lf\n", double1, double2);

    Printf("char1 = %c, char2 = %c\n", char1, char2);

    Return 0;

}
```