

COP 3502 Section 4 Quiz #1 Version C (SLMP, Dynamic Memory Allocation) Solution

1) (5 pts) Write a single line of code which dynamically allocates space for an array of size 1,000,000 each capable of storing a single variable of type `struct student`. Name the pointer variable `list`.

```
struct student* list = calloc(1000000, sizeof(struct student)); // OR
struct student* list = malloc(1000000*sizeof(struct student));
```

Grading: 1 pt LHS, 1 pt malloc or calloc, 1 pt # of terms, 1 pt sizeof, 1 pt matching parameters properly to function choice

2) (10 pts) Complete the function below so that it returns 1 if there exist two different numbers in the array `list` that multiply to exactly `target`, and 0 otherwise.. In order to earn full credit, the function must run in $O(n)$ time. You may assume that the numbers in the array are all positive and less than 30,000.

```
// Pre-condition: list is length n, sorted, with unique values.
// Post-condition: Returns 1 if two different values in list multiply
//                 to exactly target, and 0 if no such pair exists.
int mult_to_target(int list[], int n, int target) {

    int i = 0, j = n-1;

    while ( i < j ) { // 1 pt

        if (list[i]*list[j] < target) // 2 pts
            i++; // 1 pt

        else if (list[i]*list[j] > target) // 2 pts
            j--; // 1 pt

        else // 1 pt
            return 1; // 1 pt
    }

    return 0; // 1 pt
}
```

3) (12 pts) Write a function that takes in a string, `word`, of uppercase letters, and an integer, `n`, where it's guaranteed that the length of the string is exactly n^2 , and returns an array of strings (`char**`), where the first string is the first `n` characters of `word`, the next string is the next `n` characters of `word`, and so on. Make sure each string in the array of strings has the null character (`\0`) properly placed in the last slot and that each of the strings is dynamically allocated to the proper length. For example if `word = "COMPUTERSCIENCEI"` the function should return a pointer to an array storing `{"COMP", "UTER", "SCIE", "NCEI"}`.

```
char** make2D(char* word, int n) {

    char** res = calloc(n, sizeof(char*));           // 3 pts

    for (int i=0; i<n; i++) {                       // 1 pt

        res[i] = calloc(n+1, sizeof(char));         // 2 pts

        for (int j=0; j<n; j++)                    // 1 pt
            res[i][j] = word[n*i+j];              // 3 pts
        res[i][n] = '\0';                          // 1 pt
    }

    return res;                                     // 1 pt

}
```

4) (8 pts) Complete the segment of code below so that it allocates an array of `n` integer arrays, such that the first array (in index 0) has 1 element, the second array has 3 elements, the third array has 5 elements, and so on. (The last array will have $2n - 1$ integers in it. Make sure each element in each array is set to 0.

```
int n;
scanf("%d", &n);

int** array = calloc(n, sizeof(int*));             // 3 pts
for (int i=0; i<n; i++)                           // 2 pts
    array[i] = calloc(2*i+1, sizeof(int));         // 3 pts
```