

EEE101 C Programming and Software Engineering

Solutions to Lab Practice 9

Exercise 1

1. The program first asks the user to input the number of double entries to allocate (by means of the statement `malloc()`).
2. If the memory is not sufficient to allocate them, then the program exits (see the first `if` statement in the program).
3. Otherwise, the program asks the user to fill up the double entries into an array (see the `while` loop, also with the possibility to break the loop when the entry is the character 'q').
4. Finally, the program does some printings and frees the memories which have been previously allocated.

Exercise 2

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

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

```
void display(int **a, int cols, int rows);
```

```
void doublevalue(int **a, int cols, int rows);
```

```
int main(){
```

```
int **marray;
```

```
int x, y, rows=3, cols=5;
```

```
marray=(int **)malloc(cols * sizeof(int *)); /*allocate memory for the marry columns*/
```

```
if(marray==NULL){
```

```
    free(marray);
```

```
    printf("Memory allocation failed while allocating for marray[].\n");
```

```
    exit(0);
```

```
}
```

```
for(x = 0; x < cols; x++){ /*allocate memory for rows of marry in each column*/
```

```
    marray[x]=(int *)malloc(rows * sizeof(int));
```

```
    if(marray[x]==NULL){
```

```
        free(marray[x]);
```

```
        printf("Memory allocation failed while allocating for marray[x][].\n");
```

```
        exit(0);
```

```
    }
```

```
}
```

```
for(y = 0; y < rows; y++){ /*initialize marray to be filled with some values*/
```

```
    for(x = 0; x < cols; x++){
```

```
        marray[x][y] = x+y;
```

```
    }
```

```
}
```

```
printf("Print all values:\n");
display(marray,cols,rows);
```

```
doublevalue(marray,cols,rows);
printf("Double the values:\n");
display(marray,cols,rows);
```

```
for(x = 0; x < cols; x++)                /*to free the memory*/
    free(marray[x]);
```

```
free(marray);
}
```

```
void display(int **a, int cols, int rows){    /*function used to print the values*/
    int i, j;
    for(i = 0; i < cols; i++){
        for(j = 0; j < rows; j++){
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
}
```

```
void doublevalue(int **a, int cols, int rows){    /*function used to double the values*/
    int i, j;
    for(i = 0; i < cols; i++){
        for(j = 0; j < rows; j++){
            a[i][j] *= 2;
        }
    }
}
```

Exercise 3

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

int main(){
    int number, i, p=2, pcount=0;
    int *primes;

    printf("Enter the range: ");
    scanf("%d", &number);

    primes=(int *)malloc((number+1) * sizeof(int));
    if(primes==NULL){
        printf("The memory is insufficient. So the user has to formulate a lesser request and try it again");
        exit(0);
    }
    for(i=0; i<=number; i++)
        primes[i] = i;
    primes[number+1]=-1;
    primes[0]=-2;
    primes[1]=-2;

    while(p*p < number ){
        for(i=p; primes[i]!=-1; i++){
            if(primes[i]%p==0 && primes[i]!=p){
                primes[i]=-2;
            }
        }
        p++;
        while(primes[p]==-2)
            p++;
    }
    printf("Between 1 and %d, the prime numbers are:\n", number);
    p=0;
    while(primes[p] != -1){
        if(primes[p] != -2){
            printf("%d\t", primes[p]);
            pcount++;
        }
        p++;
    }

    printf("\nThe total count of prime numbers is: %d\n", pcount);
    free(primes);
    return 0;
}
```