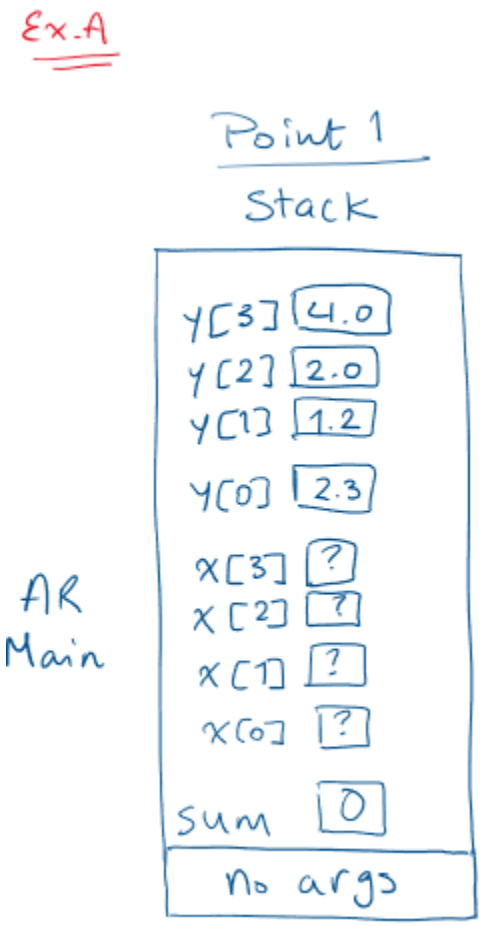
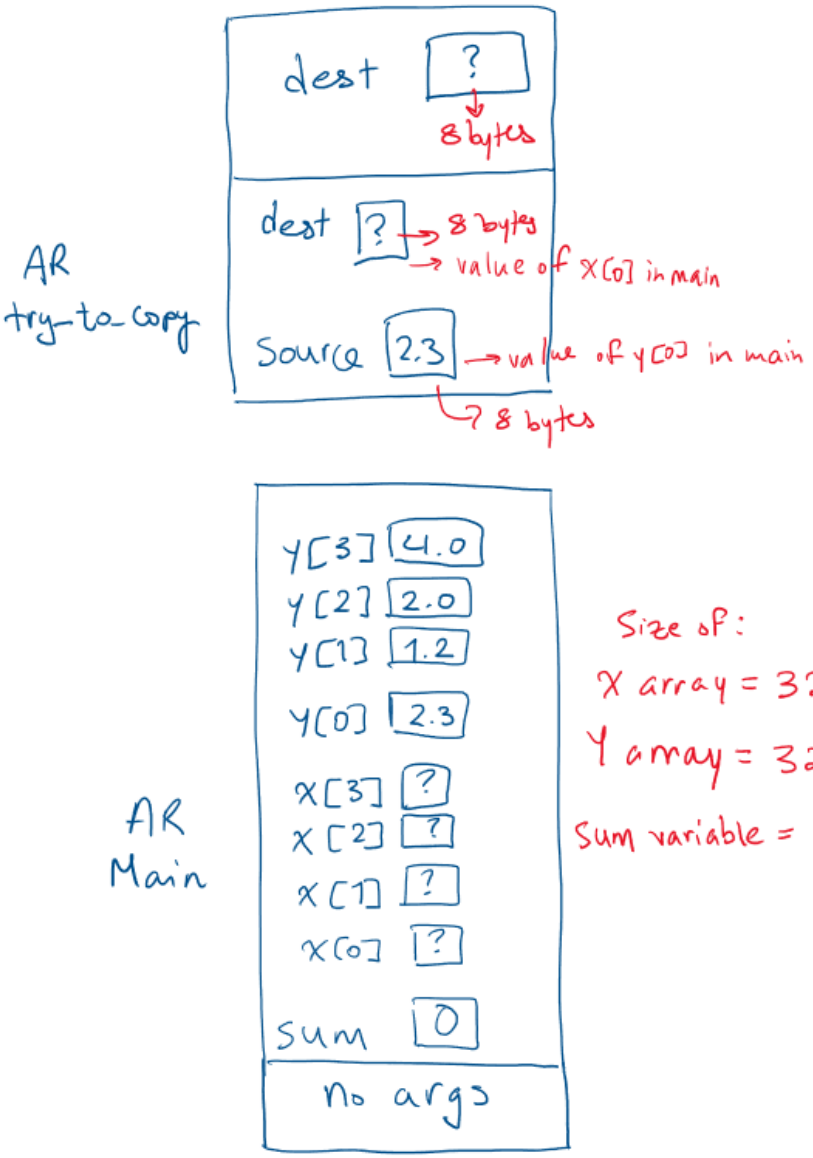
Course: Programming Fundamental – ENSF 337
Lab #: Lab 3
Instructor: M. Moussavi
Student Name: Aarushi Roy Choudhury
Lab Section: B01
Date submitted: Oct, 14 2021

Exercise A


Ex.A
Point 1
Stack
y[3] 4.0
y[2] 2.0
y[1] 1.2
y[0] 2.3
x[3] ?
x[2] ?
x[1] ?
x[0] ?
sum 0
AR
Main
no args

Point 2

Stack

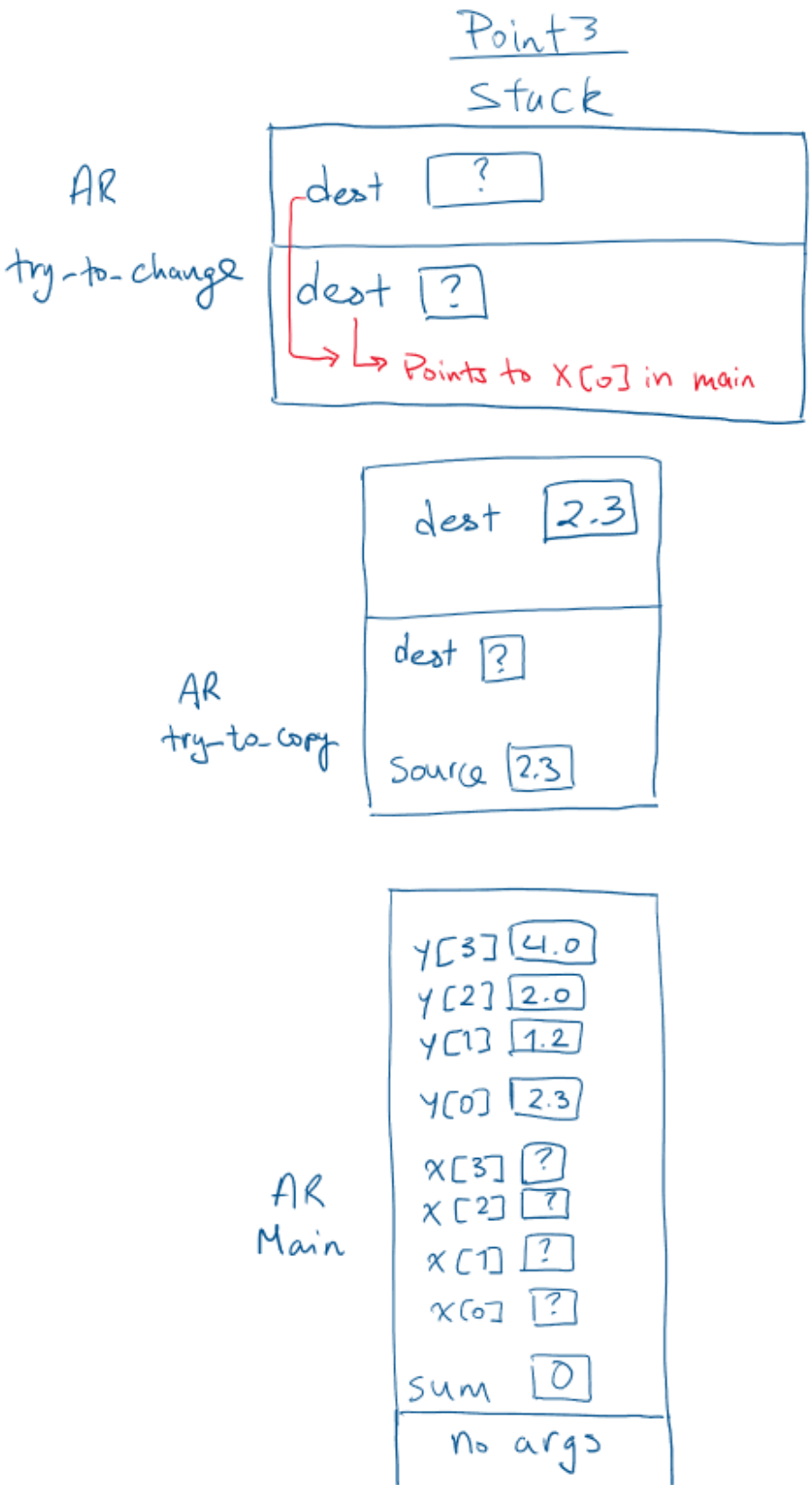


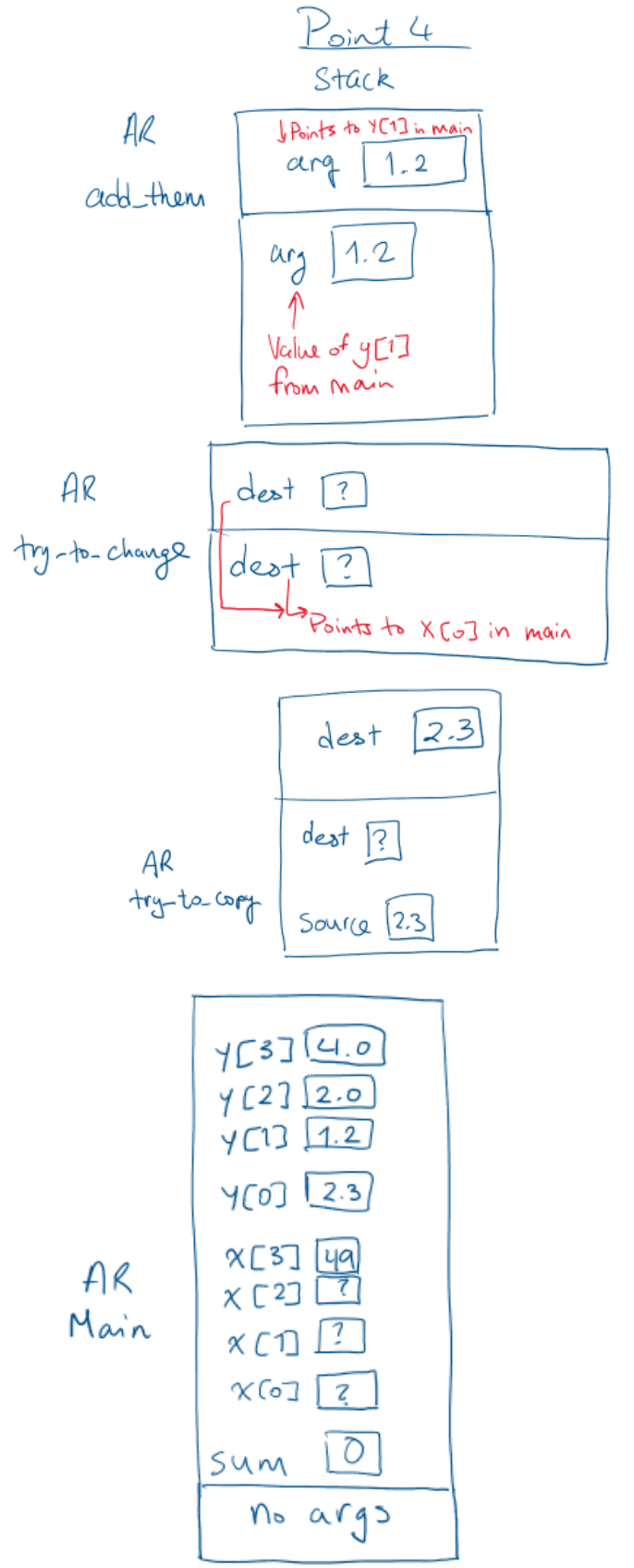
Size of:

X array = 32 bytes

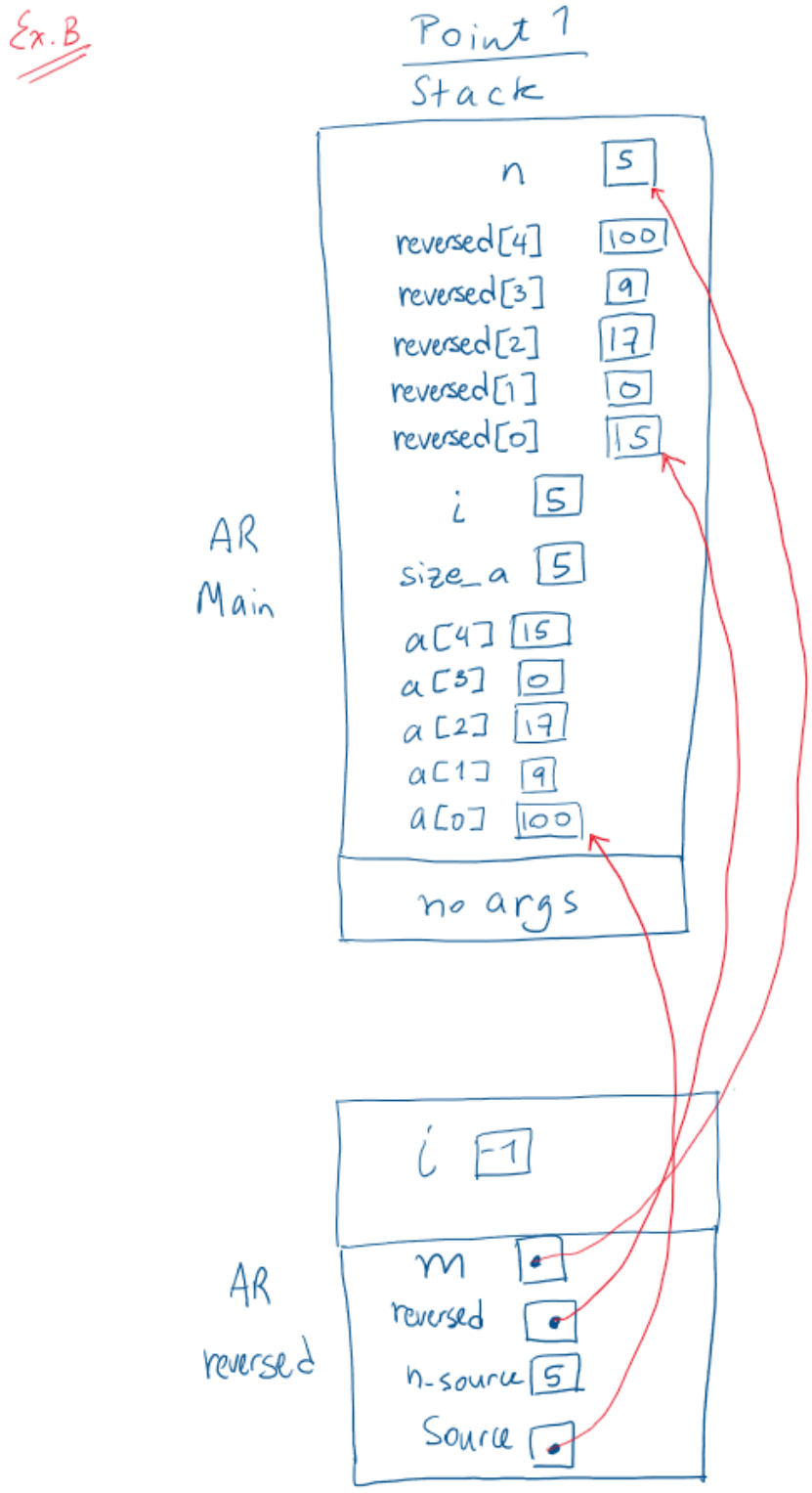
Y array = 32 bytes

Sum variable = 8 bytes


Point3
Stack
AR
try-to-change
dest
?
dest
?
Points to X[0] in main
dest
2.3
dest
?
AR
try-to-copy
Source
2.3
Y[3]
4.0
Y[2]
2.0
Y[1]
1.2
Y[0]
2.3
AR
Main
X[3]
?
X[2]
?
X[1]
?
X[0]
?
sum
0
no args

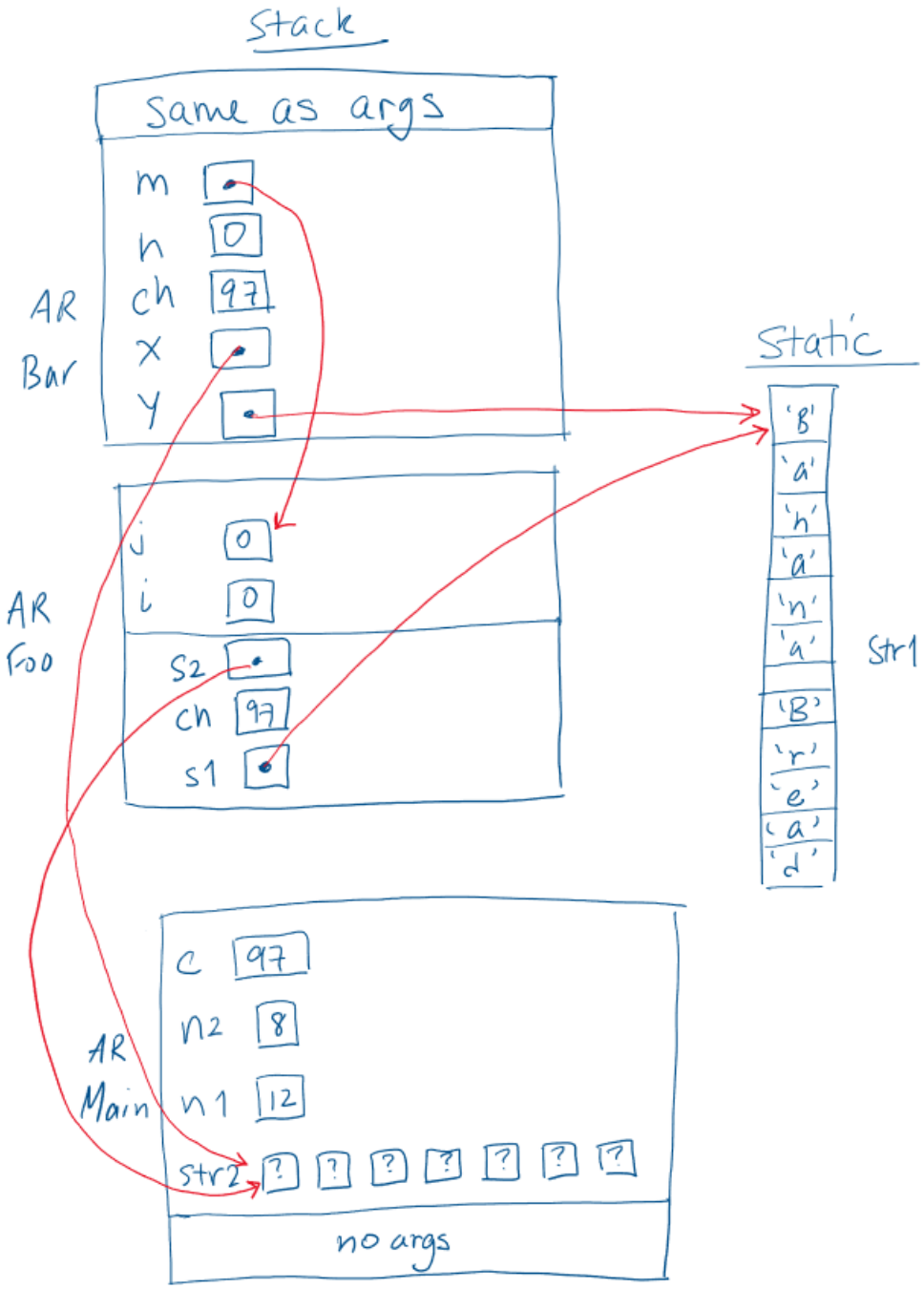

Point 4
Stack
AR
add_them
Points to Y[1] in main
arg
1.2
arg
1.2
Value of y[1]
from main
AR
try_to_change
dest
?
dest
?
Points to X[0] in main
dest
2.3
dest
?
AR
try_to_copy
Source
2.3
AR
Main
Y[3]
4.0
Y[2]
2.0
Y[1]
1.2
Y[0]
2.3
X[3]
49
X[2]
?
X[1]
?
X[0]
?
sum
0
No args

Exercise B


Ex. B
Point 1
Stack
n
5
reversed[4]
100
reversed[3]
9
reversed[2]
17
reversed[1]
0
reversed[0]
15
i
5
AR
Main
size_a
5
a[4]
15
a[3]
0
a[2]
17
a[1]
9
a[0]
100
no args
i
-1
m
reversed
n-source
5
Source
AR
reversed

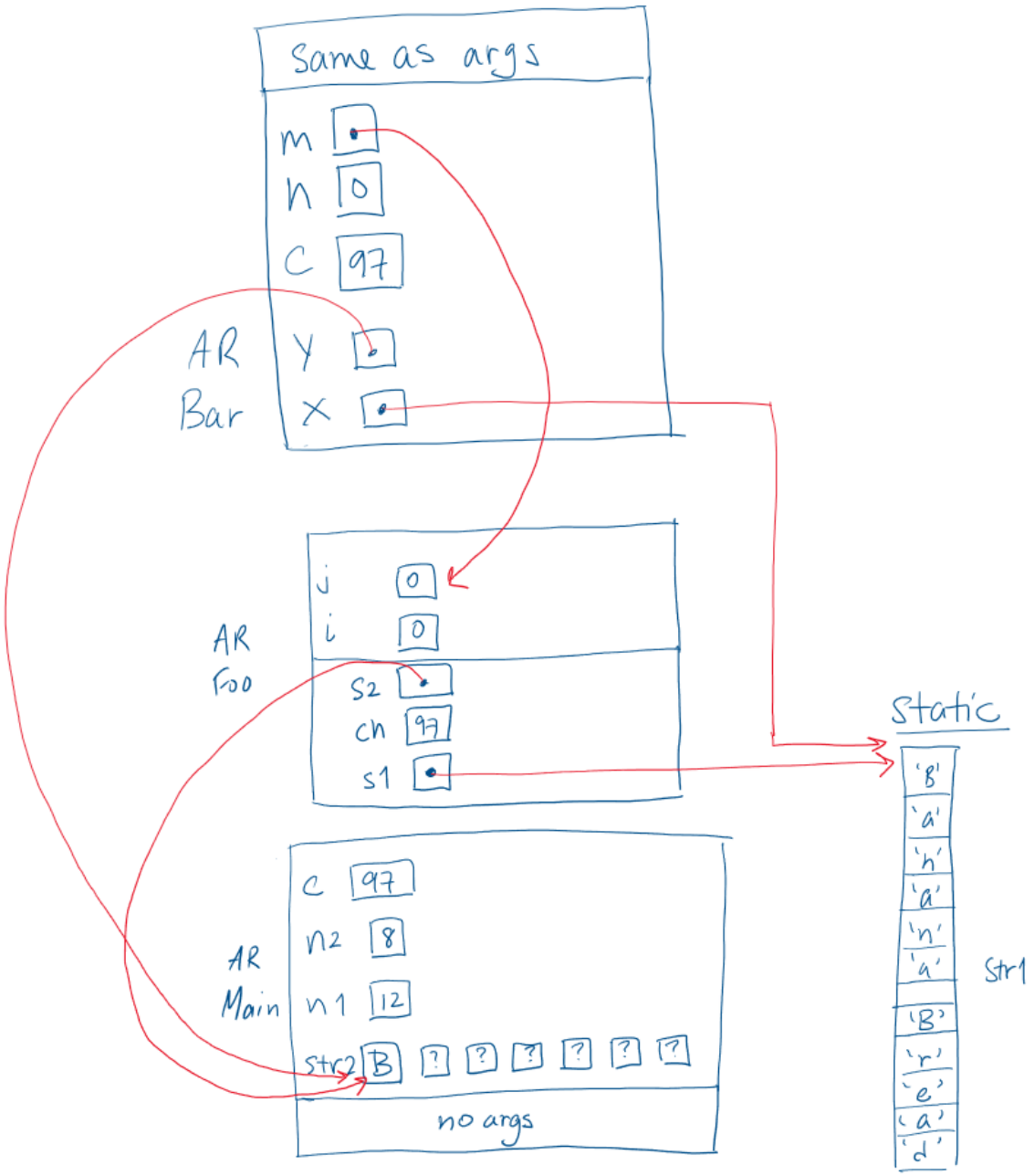
Exercise C

Point 2



Point 1

Stack



Exercise D

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

void pascal_triangle(int n);
/* REQUIRES: n > 0 and n <= 20
 PROMISES: displays a pascal_triangle. the first 5 line of the function's output
 should have the following format:
 row 0:  1
 row 1:  1   1
 row 2:  1   2   1
 row 3:  1   3   3   1
 row 4:  1   4   6   4   1
 */

int main() {
    int nrow;
    // These are ALL of the variables you need!
    printf("Enter the number of rows (Max 20): ");
    scanf("%d", &nrow);
    if(nrow <= 0 || nrow > 20) {
        printf("Error: the maximum number of rows can be 20.\n");
        exit(1);
    }

    pascal_triangle(nrow);
    return 0;
}

void pascal_triangle(int n) {
   int arr[n][n];
   int i=0,j=0;

   for(i=0;i<n;i++){
       for(j=0;j<=i;j++){
            if(j==0 || j==i)
                arr[i][j]=1;
           else
               arr[i][j]=arr[i-1][j-1]+arr[i-1][j];
       }
   }

   for(i=0;i<n;i++){
       printf("row %d:",i );
```

```
        for(j=0;j<=i;j++){
            printf("   %d",arr[i][j] );
        }
        printf("\n");
    }
}
```

```
Enter the number of rows (Max 20): 6
row 0:   1
row 1:   1   1
row 2:   1   2   1
row 3:   1   3   3   1
row 4:   1   4   6   4   1
row 5:   1   5   10   10   5   1
PS C:\Users\Aarus\Desktop\ENSF 337\Lab 3>
```

Exercise E

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

int substring(const char *s1, const char *s2);
/* REQUIRES
 * s1 and s2 are valid C-string terminated with '\0';
 * PROMISES
 * returns one if s2 is a substring of s1). Otherwise returns zero.
 */

void select_negatives(const int *source, int n_source,
                      int* negatives_only, int* number_of_negatives);
/* REQUIRES
 *   n_source >= 0.
 *   Elements source[0], source[1], ..., source[n_source - 1] exist.
 *   Elements negatives_only[0], negatives_only[1], ..., negatives_only[n_source
- 1] exist.
 * PROMISES
 *   number_of_negatives == number of negative values in source[0], ...,
source[n_source - 1].
 *   negatives_only[0], ..., negatives_only[number_of_negatives - 1] contain
those negative values, in
 *   the same order as in the source array.                              */
```

```c
int main(void)
{
    char s[] = "Knock knock! Who's there?";
    int a[] = { -10, 9, -17, 0, -15 };
    int size_a;
    int i;
    int negative[5];
    int n_negative;

    size_a = sizeof(a) / sizeof(a[0]);

    printf("a has %d elements:", size_a);
    for (i = 0; i < size_a; i++)
        printf("  %d", a[i]);
    printf("\n");
    select_negatives(a, size_a, negative, &n_negative);
    printf("\nnegative elements from array a are as follows:");
    for (i = 0; i < n_negative; i++)
        printf("  %d", negative[i]);
    printf("\n");

    printf("\nNow testing substring function....\n");
    printf("Answer must be 1. substring function returned: %d\n" , substring(s,
"Who"));
    printf("Answer must be 0. substring function returned: %d\n" , substring(s,
"knowk"));
    printf("Answer must be 1. substring function returned: %d\n" , substring(s,
"knock"));
    printf("Answer must be 0. substring function returned: %d\n" , substring(s,
""));
    printf("Answer must be 1. substring function returned: %d\n" , substring(s,
"ck! Who's"));
    printf("Answer must be 0. substring function returned: %d\n" , substring(s,
"ck!Who's"));
    return 0;
}

int substring(const char *s1, const char* s2)
{
    int i,j,k;

    for(i=0;s1[i] !='\0';i++){
        for(j=i,k=0;s2[k]!='\0'&& s1[j]==s2[k];j++,k++){
            ;
```

```
        }
        if(k >0 && s2[k]=='\0')
          return 1;
    }
     return 0;
}


void select_negatives(const int *source, int n_source,
                      int* negatives_only, int* number_of_negatives)
{

    int i;
    *number_of_negatives = 0;
     int j=0;

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

        if(source[i]<0){
            negatives_only[j]=source[i];
            j++;
        }
    }
    *number_of_negatives = j;
    return ;
}
```

```
a has 5 elements:  -10  9  -17  0  -15

negative elements from array a are as follows:  -10  -17  -15

Now testing substring function....
Answer must be 1. substring function returned: 1
Answer must be 0. substring function returned: 0
Answer must be 1. substring function returned: 1
Answer must be 0. substring function returned: 0
Answer must be 1. substring function returned: 1
Answer must be 0. substring function returned: 0
PS C:\Users\Aarus\Desktop\ENSF 337\Lab 3>
```

Exercise F

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define SIZE 100

int is_palindrome (const char *str);
/* REQUIRES: str is pointer to a valid C string.
 * PROMISES: the return value is 1 if the string a is palindrome.*/

void strip_out(char *str);
/* REQUIRES: str points to a valid C string terminated with '\0'.
 * PROMISES: strips out any non-alphanumerical characters in str*/

int main(void)
{
    int p =0;
    char str[SIZE], temp[SIZE];

    fgets(str, SIZE, stdin);

    if (str[strlen(str) - 1] == '\n')
        str[strlen(str) - 1] = '\0';

    strcpy(temp,str);

    while(strcmp(str, "done") !=0)
    {

#if 1
        strip_out(str);

        p = is_palindrome(str);
#endif

        if(!p)
            printf("\n \"%s\" is not a palindrome.", temp);
        else
            printf("\n \"%s\" is a palindrome.", temp);

        fgets(str, SIZE, stdin);

        /* Remove end-of-line character if there is one in str.*/
        if(str[strlen(str) - 1] == '\n')
```

```
            str[strlen(str) - 1]= '\0';
        strcpy(temp, str);
    }

    return 0;
}
int is_palindrome(const char *str){

    int i = 0;
    int j = strlen(str) - 1;
    char x;
    char y;

    while(j > i){
        x = str[i];
        y = str[j];

        if(isupper(x)){
            x = tolower(x);
        }

        if(isupper(y)){
            y = tolower(y);
        }

        if(x != y){
            return 0;
        }

        ++i;
        --j;
    }
    return 1;
}


void strip_out(char *str){

    char *p;
    char copy[100];
    int len = 0;
    for (p = str; *p != '\0'; p++) {

        if(isalnum(*p)){
           copy[len] = *p;
```

```
            ++len;
        }
    }
    copy[len] = '\0';
    strcpy(str, copy);

}
```

```
C:\Users\Aarus\Desktop\ENSF 337\Lab 3>gcc palindrome.c

C:\Users\Aarus\Desktop\ENSF 337\Lab 3>a.exe < palindrome.txt

 "Radar" is a palindrome.
 "Madam I'm Adam" is a palindrome.
 "Alfalfa" is not a palindrome.
 "He maps spam, eh?" is a palindrome.
 "I did, did I?" is a palindrome.
 "       I prefer pi." is a palindrome.
 "Ed is on no side" is a palindrome.
 "Am I loco, Lima?" is a palindrome.
 "         Bar crab." is a palindrome.
 "A war at Tarawa." is a palindrome.
 "Ah, Satan sees Natasha" is a palindrome.
 "     Borrow or rob?" is a palindrome.
 "233332" is a palindrome.
 "324556" is not a palindrome.
 "Hello world!!" is not a palindrome.
 "     Avon sees nova  " is a palindrome.
 "Can I attain a 'C'?" is a palindrome.
 "Sept 29, 2005." is not a palindrome.
 "Delia failed." is a palindrome.
 "Draw nine men $$  inward" is a palindrome.
C:\Users\Aarus\Desktop\ENSF 337\Lab 3>
```