

Name: Muhammad Abbas ZaidiSection: B-2Rollno: 150415

Question 1. (10 points) Read the following C program and write the output for the five separate input values.

```

1 #include <stdio.h>
2
3 void permute(int* arr, int n) {
4     int i;
5     for(i=1; i<n; i++) {
6         if(arr[i] >= arr[i-1])
7             continue;
8         int j;
9         for(j=0; j<i; j++) //place arr[i] correctly
10             if(arr[j] > arr[i]) {
11                 int k=j;
12                 int prev = arr[k];
13                 arr[k] = arr[i];
14                 for(k=j+1; k<=i; k++) {
15                     int curr = arr[k];
16                     arr[k] = prev;
17                     prev = curr;
18                 }
19                 break;
20             }
21     }
22 }
23
24 int compute(int *A, int n) {
25     int i;
26     permute(A, n);
27     for(i=n-1; i>=0; i--)
28         if(*(A+i) < n-i) break;
29     return n-1-i;
30 }
31
32 int main() {
33     int n, i, A[5];
34     scanf("%d", &n);
35     for(i=0; i<n; i++)
36         scanf("%d", A+i);
37     printf("%d\n", compute(A, n));
38     return 0;
39 }

```

When $a[i] < a[i-1]$ we move down $A[i] < n-i$

Handwritten notes and diagrams illustrating the execution of the program for input $n=5$ and array $A = [1, 2, 3, 4, 5]$.

Diagram 1: Initial state of array A and variable i .

1	2	3	4	5
---	---	---	---	---

$i = 5$

Diagram 2: Array A after the first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 4$

Diagram 3: Array A after the second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 3$

Diagram 4: Array A after the third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 2$

Diagram 5: Array A after the fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 1$

Diagram 6: Array A after the fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 7: Array A after the sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 8: Array A after the seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 9: Array A after the eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 10: Array A after the ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 11: Array A after the tenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 12: Array A after the eleventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 13: Array A after the twelfth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 14: Array A after the thirteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 15: Array A after the fourteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 16: Array A after the fifteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 17: Array A after the sixteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 18: Array A after the seventeenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 19: Array A after the eighteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 20: Array A after the nineteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 21: Array A after the twentieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 22: Array A after the twenty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 23: Array A after the twenty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 24: Array A after the twenty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 25: Array A after the twenty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 26: Array A after the twenty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 27: Array A after the twenty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 28: Array A after the twenty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 29: Array A after the twenty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 30: Array A after the twenty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 31: Array A after the thirtieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 32: Array A after the thirty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 33: Array A after the thirty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 34: Array A after the thirty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 35: Array A after the thirty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 36: Array A after the thirty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 37: Array A after the thirty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 38: Array A after the thirty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 39: Array A after the thirty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 40: Array A after the thirty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 41: Array A after the fortieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 42: Array A after the forty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 43: Array A after the forty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 44: Array A after the forty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 45: Array A after the forty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 46: Array A after the forty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 47: Array A after the forty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 48: Array A after the forty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 49: Array A after the forty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 50: Array A after the forty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 51: Array A after the fiftieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 52: Array A after the fifty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 53: Array A after the fifty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 54: Array A after the fifty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 55: Array A after the fifty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 56: Array A after the fifty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 57: Array A after the fifty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 58: Array A after the fifty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 59: Array A after the fifty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 60: Array A after the fifty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 61: Array A after the sixtieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 62: Array A after the sixty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 63: Array A after the sixty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 64: Array A after the sixty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 65: Array A after the sixty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 66: Array A after the sixty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 67: Array A after the sixty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 68: Array A after the sixty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 69: Array A after the sixty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 70: Array A after the sixty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 71: Array A after the seventieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 72: Array A after the seventy-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 73: Array A after the seventy-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 74: Array A after the seventy-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 75: Array A after the seventy-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 76: Array A after the seventy-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 77: Array A after the seventy-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 78: Array A after the seventy-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 79: Array A after the seventy-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 80: Array A after the seventy-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 81: Array A after the eightieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 82: Array A after the eighty-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 83: Array A after the eighty-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 84: Array A after the eighty-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 85: Array A after the eighty-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 86: Array A after the eighty-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 87: Array A after the eighty-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 88: Array A after the eighty-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 89: Array A after the eighty-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 90: Array A after the eighty-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 91: Array A after the ninetieth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 92: Array A after the ninety-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 93: Array A after the ninety-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 94: Array A after the ninety-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 95: Array A after the ninety-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 96: Array A after the ninety-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 97: Array A after the ninety-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 98: Array A after the ninety-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 99: Array A after the ninety-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 100: Array A after the ninety-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 101: Array A after the hundredth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 102: Array A after the hundred-first iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 103: Array A after the hundred-second iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 104: Array A after the hundred-third iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 105: Array A after the hundred-fourth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 106: Array A after the hundred-fifth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 107: Array A after the hundred-sixth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 108: Array A after the hundred-seventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 109: Array A after the hundred-eighth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 110: Array A after the hundred-ninth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 111: Array A after the hundred-tenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 112: Array A after the hundred-eleventh iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 113: Array A after the hundred-twelfth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 114: Array A after the hundred-thirteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 115: Array A after the hundred-fourteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 116: Array A after the hundred-fifteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

$i = 0$

Diagram 117: Array A after the hundred-sixteenth iteration of the `permute` function.

1	2	3	4	5
---	---	---	---	---

0 ABAS ZAIDI

Section:

B-2

Roll

in the output in the box.

Input	Output
2 -1 -2	1 X
3 3 0 2	2 ✓
4 1 0 0 0	1 00
4 9 10 8 3	4 X
4 10 4 6 5	4 ✓

Name: MOHID ABBAS ZAIDI

Section: 3-2

Rollno: 150415

Question 2. (10 points) Complete the following program which does the following. The program takes as input a positive integer n followed by a sequence of n -numbers which form an arithmetic progression with exactly one number *missing* in the arithmetic progression. The program should output the missing number.

For example, if $n = 4$ and the sequence of numbers is 2 5 11 14 then the program should output 8. (Explanation: It is an arithmetic progression with start=2 and difference=3.)

```

1 #include <stdio.h>
2
3 int findAP(int* A, int lo, int hi, int diff) {
4
5     int m;
6     m = lo + hi / 2;
7
8     if ((a[m+1] - a[m]) != diff) ✓
9         return (*(A+m) + diff);
10
11     if (m > 0 && (a[m] - a[m-1]) != diff) ✓
12         return (*(A+m-1) + diff);
13
14     if (a[m] - a[0] = m * diff) ✓
15         return findAP(A, m+1, hi, diff);
16
17     return findAP(A, lo, m-1, diff); ✓
18 }
19
20 int main() {
21     int* a;
22     int n, i, diff;
23     scanf("%d", &n);
24     a = (int*) malloc(n*sizeof(int));
25     for(i=0; i<n; i++)
26         scanf("%d", a+i);
27
28     diff = (a[n-1] - a[0]) / n;
29
30     printf("Missing element: %d\n", findAP(a, 0, n-1, diff));
31     return 0;
32 }

```

$$\begin{aligned}
 (a[m+1] - a[m]) &= \text{diff} \\
 (a[m] - a[m-1]) &= \text{diff}
 \end{aligned}$$

Question 3. (10 points) Read the following C program and figure out the output for the five separate input values.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int func(int a, int inc, int b) {
5     int p, n, r;
6
7     if (abs(a) > b) return 1000;
8     if (a == b) return inc;
9
10    p = func(a+inc+1, inc+1, b);
11    n = func(a-inc-1, inc+1, b);
12
13    if (p < n) r=p;
14    else r=n;
15
16    return r;
17 }
18
19 int main() {
20     int n;
21     scanf("%d",&n);
22     printf("%d\n", func(0, 0, n));
23     return 0;
24 }

```

Neatly fill in the output in the box.

Input	Output
1	1 ✓
4	3 ✓
5	5 ✓
6	3 ✓
9	5 ✓

$$0, 0, 1$$

$$p = f(1, 1, 1) = 1$$

$$n = f(-1, 1, 1) = 2$$

$$p = f(1, 2, 1) \rightarrow 2$$

$$n = f(-3, 2, 1) \rightarrow 1000$$

$$f(0, 0, 4) \rightarrow f(1, 1, 4) \rightarrow f(-1, 1, 4)$$

$$(0, 0, 4) \rightarrow (1, 1, 4) \rightarrow (3, 2, 4) \rightarrow (-1, 2, 4)$$

$$(2, 3, 4) \rightarrow (-4, 3, 4)$$

$$(6, 4, 4) \rightarrow (-2, 4)$$

$$(1, 5) \rightarrow 1000 \rightarrow 100$$

$$0, 1 \rightarrow 3, 2 \rightarrow 6, 3 \rightarrow 1000 \rightarrow 7, 5 \times$$

$$-4, 1 \rightarrow (2, 2) \rightarrow (6, 3) \rightarrow (10, 4)$$

$$\rightarrow (15, 4) \rightarrow (0, 4)$$