

IOT Assignment

AI Department

July 26, 2023

Contents

| | | |
|----------|-----------------------|----------|
| 1 | Question 1 | 2 |
| 2 | Question 2 | 2 |
| 3 | Question 3 | 3 |
| 4 | Question 4 | 4 |
| 5 | Question 5 | 5 |
| 6 | Question 6 | 5 |
| 7 | Bonus Question | 6 |

1 Question 1

Algorithm 1 Question 1

```
1: function SUM(arr, length)
2:    $sum \leftarrow 0$ 
3:   for  $i \leftarrow 0$  to  $length - 1$  do
4:      $sum \leftarrow sum + arr[i]$ 
5:   return sum
6: Output :
7: The total sum of this array: 75
```

2 Question 2

Algorithm 2 Question 2

```
1: function CALC_AVG(numbers, length)
2:    $sum \leftarrow 0$ 
3:   for  $i \leftarrow 0$  to  $length - 1$  do
4:      $sum \leftarrow sum + numbers[i]$ 
5:   return  $double(sum)/double(length)$ 
6: Output :
7: Please fill the array
8: Element Number-1: 1
9: Element Number-2: 2
10: Element Number-3: 3
11: Element Number-4: 4
12: Element Number-5: 5
13: Average value of the array: 3
```

3 Question 3

Algorithm 3 Question 3

```
1: function FILL_ARRAY(arr, length)
2:   for  $i \leftarrow 0$  to  $length - 1$  do
3:     print "Element Number-" +  $(i + 1)$  + ": "
4:     value  $\leftarrow$  read()
5:      $arr[i] \leftarrow$  value

6: Output :
7: Enter size of the array: 10
8: Element Number-1: 1
9: Element Number-2: 3
10: Element Number-3: 5
11: Element Number-4: 7
12: Element Number-5: 9
13: Element Number-6: 2
14: Element Number-7: 4
15: Element Number-8: 6
16: Element Number-9: 8
17: Element Number-10: 10
```

Algorithm 4 Question 3

```
1: function SWAP_(x, y)
2:   temp  $\leftarrow$   $*x$ 
3:    $*x \leftarrow *y$ 
4:    $*y \leftarrow$  temp
```

Algorithm 5 Question 3

```
1: function FIND_MAX_PRODUCT(arr, length)
2:   max_product  $\leftarrow$  1
3:   for i  $\leftarrow$  0 to length - 1 do
4:     for j  $\leftarrow$  0 to length - 1 do
5:       if arr[j] > arr[j + 1] then
6:         SWAP_(&arr[j], &arr[j+1])
7:   print "Max 3 numbers in this array: "
8:   for i  $\leftarrow$  0 to 2 do
9:     print arr[length - 1 - i]
10:    max_product  $\leftarrow$  max_product  $\times$  arr[length - 1 - i]
11:   return max_product
12: Output :
13: Max 3 numbers in this array: 10 9 8
14: Maximum Product = 720
```

4 Question 4

Algorithm 6 Question 4

```
1: function COUNT_VOWELS(str)
2:   the_count  $\leftarrow$  0
3:   while *str  $\neq$  '\0' do
4:     ch  $\leftarrow$  *str
5:     ch  $\leftarrow$  tolower(ch)
6:     if ch = 'a' or ch = 'e' or ch = 'i' or ch = 'o' or ch = 'u' then
7:       the_count  $\leftarrow$  the_count + 1
8:     str  $\leftarrow$  str + 1
9:   return the_count
10: Output :
11: Enter a string: OmarFaresEhabDawoodYousry
12: Number of vowels in OmarFaresEhabDawoodYousry: 11
```

5 Question 5

Algorithm 7 Question 5

```
1: function SUM(arr, length)
2:   sum  $\leftarrow$  0
3:   for i  $\leftarrow$  0 to length - 1 do
4:     sum  $\leftarrow$  sum + arr[i]
       $\triangleright$  Optionally, you can display the result using print or return
5:   print "The sum of the array: " + sum
6:   return sum
7: Output :
8: Please Fill The Array
9: Number-1: 1
10: Number-2: 2
11: Number-3: 3
12: Number-4: 4
13: Number-5: 5
14: Sum of elements in the array: 15
```

6 Question 6

Algorithm 8 Question 6

```
1: function FACTORIAL(n)
2:   prod  $\leftarrow$  1
3:   for i  $\leftarrow$  1 to n do
4:     prod  $\leftarrow$  prod  $\times$  i
5:   return prod
6: Output :
7: Please enter a number to get it's factorial: 5
8: Factorial: 120
```

7 Bonus Question

Algorithm 9 Merge Function

```
1: function MERGE(arr, l, m, r)
2:    $n1 \leftarrow m - l + 1$ 
3:    $n2 \leftarrow r - m$ 
4:   allocate left_arr of size  $n1$ 
5:   allocate right_arr of size  $n2$ 
6:   for  $i \leftarrow 0$  to  $n1 - 1$  do
7:      $left\_arr[i] \leftarrow arr[l + i]$ 
8:   for  $j \leftarrow 0$  to  $n2 - 1$  do
9:      $right\_arr[j] \leftarrow arr[m + 1 + j]$ 
10:   $i \leftarrow j \leftarrow 0$ 
11:   $k \leftarrow l$ 
12:  while  $i < n1$  and  $j < n2$  do
13:    if  $left\_arr[i] \leq right\_arr[j]$  then
14:       $arr[k] \leftarrow left\_arr[i]$ 
15:       $i \leftarrow i + 1$ 
16:    else
17:       $arr[k] \leftarrow right\_arr[j]$ 
18:       $j \leftarrow j + 1$ 
19:       $k \leftarrow k + 1$ 
20:  while  $i < n1$  do
21:     $arr[k] \leftarrow left\_arr[i]$ 
22:     $i \leftarrow i + 1$ 
23:     $k \leftarrow k + 1$ 
24:  while  $j < n2$  do
25:     $arr[k] \leftarrow right\_arr[j]$ 
26:     $j \leftarrow j + 1$ 
27:     $k \leftarrow k + 1$ 
28:  deallocate left_arr
29:  deallocate right_arr
```

▷ Allocate temporary arrays

▷ Deallocate the temporary arrays

Algorithm 10 Merge Function

```
1: function MERGE_SORT(arr, l, r)
2:   if  $l < r$  then  $\triangleright$  Stopping condition: array has more than one element
3:      $m \leftarrow l + \frac{r-l}{2}$ 
4:     MERGE_SORT(arr, l, m)
5:     MERGE_SORT(arr, m+1, r)
6:     MERGE(arr, l, m, r)
7: Output:
8: The random array before sorting
9: _____
10: 45 83 38 44 17 32 15 80 5 54 51 7 10 75 30 54 92 64 32 66 75
11: _____
12: the array after sorting
13: _____
14: 5 7 10 15 17 30 32 32 38 44 45 51 54 54 64 66 75 75 80 83 92
15: _____
```

Algorithm 11 Median Function

```
1: function MEDIAN(arr, n)
2:   if  $n \bmod 2 = 0$  then  $\triangleright$  To get the median in case of even
3:     return  $arr[\frac{n}{2} - 1]$ 
4:   return  $arr[\frac{n}{2}]$   $\triangleright$  To get the median in case of odd
5: Output:
6: the median of the array is : 45
7: _____
```
