

PPS8

Q1

Aim:

Simulate the MATH calculator for the following operators (+, *, \$) by using C functions.

'+' Operator does the following: Accept 2 integers as input and perform the summation of the given two numbers and return the sum as answer.

'*' Operator: Accepts 2 integers as input and multiply the given two numbers and return the product as answer.

'\$' Operator: Accept an integer as input and return the reverse of the given number.

Procedure:

Input:

Operator
Operands

Output:

Addition, multiplication or reverse of the number

Algorithm:

Step 1: Declare 'add', 'multiplication' and 'reverse' function with return type 'int' and argument 'void'

Main Function

Step 1: Read operator, op

Step 2: Use switch case for operator

Case 1 ('+'): Call function 'add' and print its return value

Case 2 ('*'): Call function 'multiplication' and print its return value

Case 3 ('\$'): Call function 'reverse' and print its return value

Default: Print error message for invalid input

Add Function

Step 1: Declare integer variables 'a', 'b' and 'sum'

Step 2: Read 'a' and 'b'

Step 3: sum = a + b

Step 4: return sum

Multiply Function

Step 1: Declare integer variables 'a', 'b' and 'prod'

Step 2: Read 'a' and 'b'

Step 3: $prod = a * b$

Step 4: return prod

Reverse Function

Step 1: Declare integer variable 'a' and 'rev'

Step 2: Read 'a'

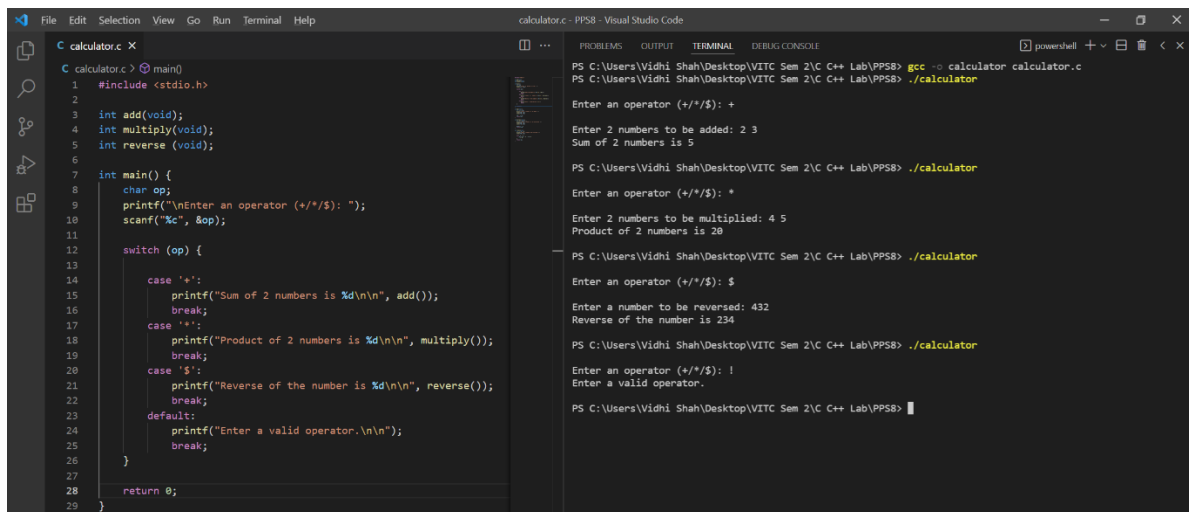
Step 3: While $n > 0$

$rev = (rev * 10) + (a \% 10)$

$a = a / 10$

Step 4: return rev

Code:



```
calculator.c X
C calculator.c > main()
1 #include <stdio.h>
2
3 int add(void);
4 int multiply(void);
5 int reverse(void);
6
7 int main() {
8     char op;
9     printf("\nEnter an operator (+/*/$): ");
10    scanf("%c", &op);
11
12    switch (op) {
13
14        case '+':
15            printf("Sum of 2 numbers is %d\n\n", add());
16            break;
17        case '*':
18            printf("Product of 2 numbers is %d\n\n", multiply());
19            break;
20        case '$':
21            printf("Reverse of the number is %d\n\n", reverse());
22            break;
23        default:
24            printf("Enter a valid operator.\n\n");
25            break;
26    }
27
28    return 0;
29 }
```

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> gcc -o calculator calculator.c
PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): +
Enter 2 numbers to be added: 2 3
Sum of 2 numbers is 5

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): *
Enter 2 numbers to be multiplied: 4 5
Product of 2 numbers is 20

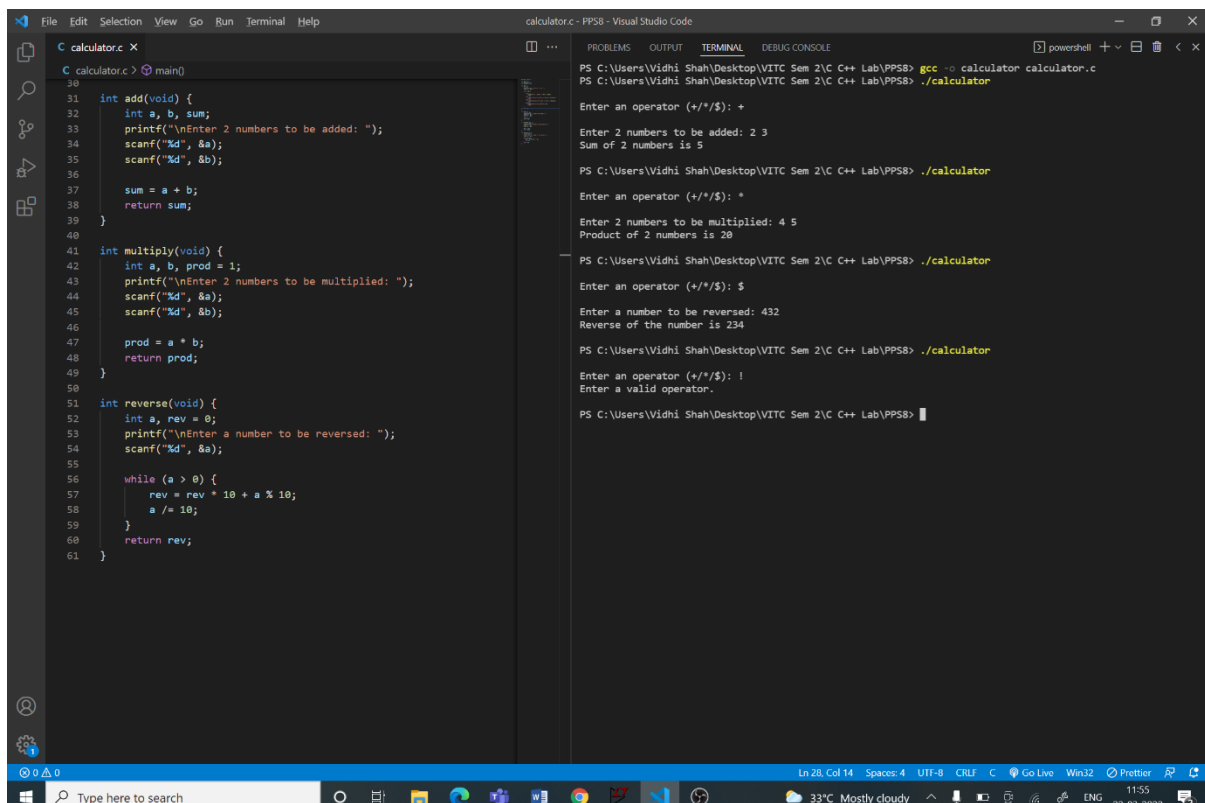
PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): \$
Enter a number to be reversed: 432
Reverse of the number is 234

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): !
Enter a valid operator.

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8>



```
calculator.c X
C calculator.c > main()
30
31 int add(void) {
32     int a, b, sum;
33     printf("\nEnter 2 numbers to be added: ");
34     scanf("%d", &a);
35     scanf("%d", &b);
36
37     sum = a + b;
38     return sum;
39 }
40
41 int multiply(void) {
42     int a, b, prod = 1;
43     printf("\nEnter 2 numbers to be multiplied: ");
44     scanf("%d", &a);
45     scanf("%d", &b);
46
47     prod = a * b;
48     return prod;
49 }
50
51 int reverse(void) {
52     int a, rev = 0;
53     printf("\nEnter a number to be reversed: ");
54     scanf("%d", &a);
55
56     while (a > 0) {
57         rev = rev * 10 + a % 10;
58         a /= 10;
59     }
60     return rev;
61 }
```

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> gcc -o calculator calculator.c
PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): +
Enter 2 numbers to be added: 2 3
Sum of 2 numbers is 5

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): *
Enter 2 numbers to be multiplied: 4 5
Product of 2 numbers is 20

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): \$
Enter a number to be reversed: 432
Reverse of the number is 234

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./calculator

Enter an operator (+/*/\$): !
Enter a valid operator.

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8>

Q2

Aim:

Given an array, write a function to reverse the first and second half of an array keeping its center element unchanged if the array elements are odd.

Procedure:

Input:

Number of elements in the array, 'n'

'Array of length 'n'

Output:

Array with 2 halves reversed separately

Algorithm:

Step 1: Declare 'reverse' function with return type 'void' and argument 'void'

Main Function

Step 1: Call function 'reverse'

Reverse Function

Step 1: Initialise integer variable 'n' and 'i'

Step 2: Read 'n'

Step 3: Initialise 2 integer arrays of size n, 'arr' and 'rev'. Let $mid = n/2$

Step 4: Read elements into the array

Step 5: For 'i' from 0 to mid (excluding mid)

Step A: $rev[mid - i - 1] = arr[i]$

Step 6: If n is odd

Step A: $rev[mid] = arr[mid]$

Step B: For 'i' from 0 to mid (excluding mid)

Step B1: $rev[n - i - 1] = arr[mid + i + 1]$

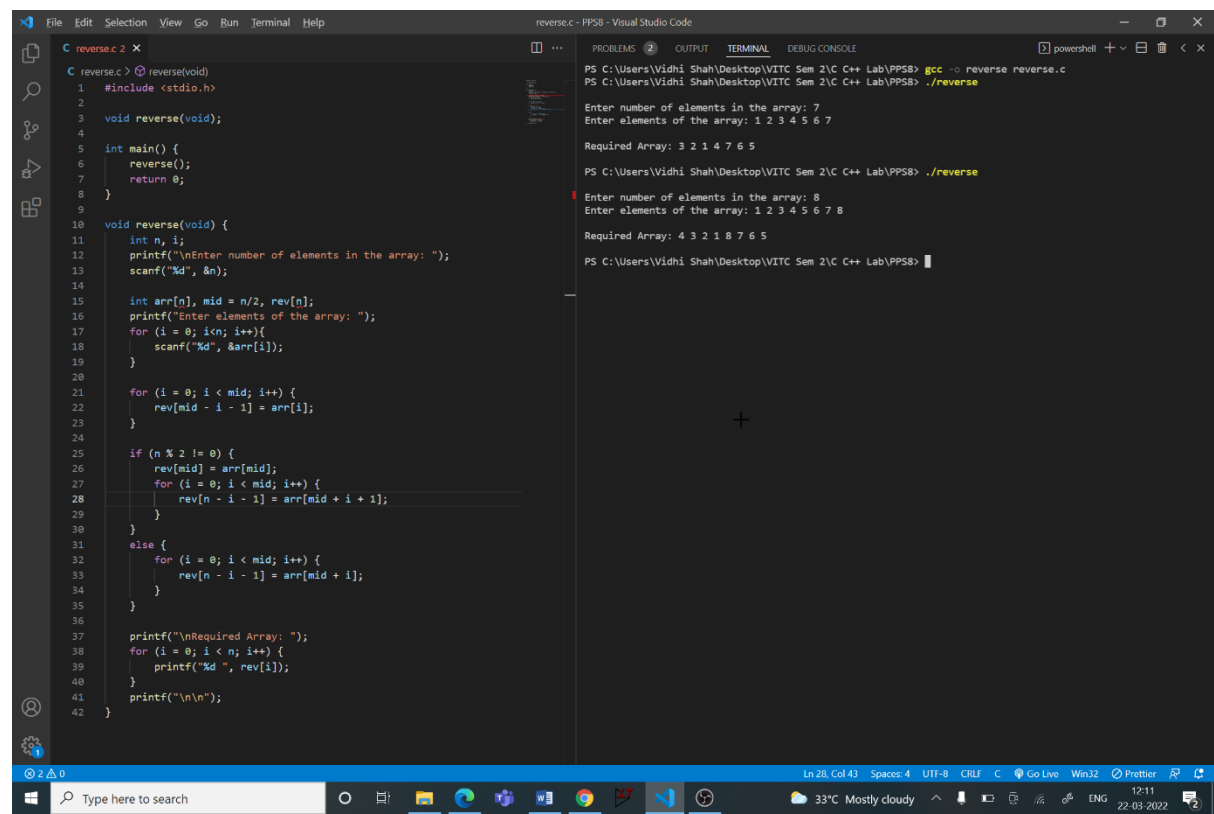
Step 6: If n is even

Step A: For 'i' from 0 to mid (excluding mid)

Step A1: $rev[n - i - 1] = arr[mid + i]$

Step 7: Display the rev array

Code:



```
reverse.c - PPSB - Visual Studio Code

C reverse.c 2 X
C reverse.c > reverse(void)
1 #include <stdio.h>
2
3 void reverse(void);
4
5 int main() {
6     reverse();
7     return 0;
8 }
9
10 void reverse(void) {
11     int n, i;
12     printf("\nEnter number of elements in the array: ");
13     scanf("%d", &n);
14
15     int arr[n], mid = n/2, rev[n];
16     printf("Enter elements of the array: ");
17     for (i = 0; i < n; i++) {
18         scanf("%d", &arr[i]);
19     }
20
21     for (i = 0; i < mid; i++) {
22         rev[mid - i - 1] = arr[i];
23     }
24
25     if (n % 2 != 0) {
26         rev[mid] = arr[mid];
27         for (i = 0; i < mid; i++) {
28             rev[n - i - 1] = arr[mid + i + 1];
29         }
30     }
31     else {
32         for (i = 0; i < mid; i++) {
33             rev[n - i - 1] = arr[mid + i];
34         }
35     }
36
37     printf("\nRequired Array: ");
38     for (i = 0; i < n; i++) {
39         printf("%d ", rev[i]);
40     }
41     printf("\n\n");
42 }
```

PROBLEMS 2 OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPSB> gcc -o reverse reverse.c

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPSB> ./reverse

Enter number of elements in the array: 7

Enter elements of the array: 1 2 3 4 5 6 7

Required Array: 3 2 1 4 7 6 5

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPSB> ./reverse

Enter number of elements in the array: 8

Enter elements of the array: 1 2 3 4 5 6 7 8

Required Array: 4 3 2 1 8 7 6 5

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPSB>

Ln 28, Col 43 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier

Type here to search 33°C Mostly cloudy 12:11 22-03-2022

Q3

Aim:

Given an array, write a function to insert a number 'x' at index 0.

Procedure:

Input:

Number of elements in the array, 'n'

Array of length 'n'

A number, 'x'

Output:

Array with 'x' at 0th index

Algorithm:

Step 1: Declare 'insert0' function with return type 'void' and argument 'void'

Main Function

Step 1: Call function 'insert0'

Insert0 Function

Step 1: Initialise integer variable 'n', 'i' and 'x'

Step 2: Read 'n'

Step 3: Initialise an integer array of size $n + 1$

Step 4: Read elements into the array

Step 5: Read 'x'

Step 6: Increment n by 1

Step 7: For 'i' from $n - 1$ to -1 (excluding -1)

Step A: $arr[i] = arr[i-1]$

Step 8: $arr[0] = x$

Step 9: Display the array

Code:

```
1 #include <stdio.h>
2
3 void insert0(void);
4
5 int main() {
6     insert0();
7     return 0;
8 }
9
10 void insert0(void) {
11     int n, i, x;
12     printf("\nEnter number of elements in the array: ");
13     scanf("%d", &n);
14
15     int arr[n+1];
16     printf("Enter elements of the array: ");
17     for (i = 0; i < n; i++){
18         scanf("%d", &arr[i]);
19     }
20
21     printf("Enter element to be inserted at index 0: ");
22     scanf("%d", &x);
23     n++;
24
25     for(i = n - 1; i > - 1; i--){
26         arr[i] = arr[i-1];
27     }
28     arr[0] = x;
29
30     printf("\nRequired Array: ");
31     for (i = 0; i < n; i++){
32         printf("%d ", arr[i]);
33     }
34     printf("\n\n");
35 }
```

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> gcc -o insert0 insert0.c

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./insert0

Enter number of elements in the array: 6

Enter elements of the array: 1 2 3 4 5 6

Enter element to be inserted at index 0: 100

Required Array: 100 1 2 3 4 5 6

PS C:\Users\Vidhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8>

Q4

Aim:

Given an array and a new position 'p', write a function to insert a number 'x' at position 'p'

Procedure:

Input:

Number of elements in the array, 'n'

Array of length 'n'

A position, 'p'

A number, 'x'

Output:

Array with 'x' at pth index

Algorithm:

Step 1: Declare 'insert' function with return type 'void' and argument 'void'

Main Function

Step 1: Call function 'insert'

Insert Function

Step 1: Initialise integer variable 'n', 'i', 'p' and 'x'

Step 2: Read 'n'

Step 3: Initialise an integer array of size n + 1

Step 4: Read elements into the array

Step 5: Read 'p' and 'x'

Step 6: Increment n by 1

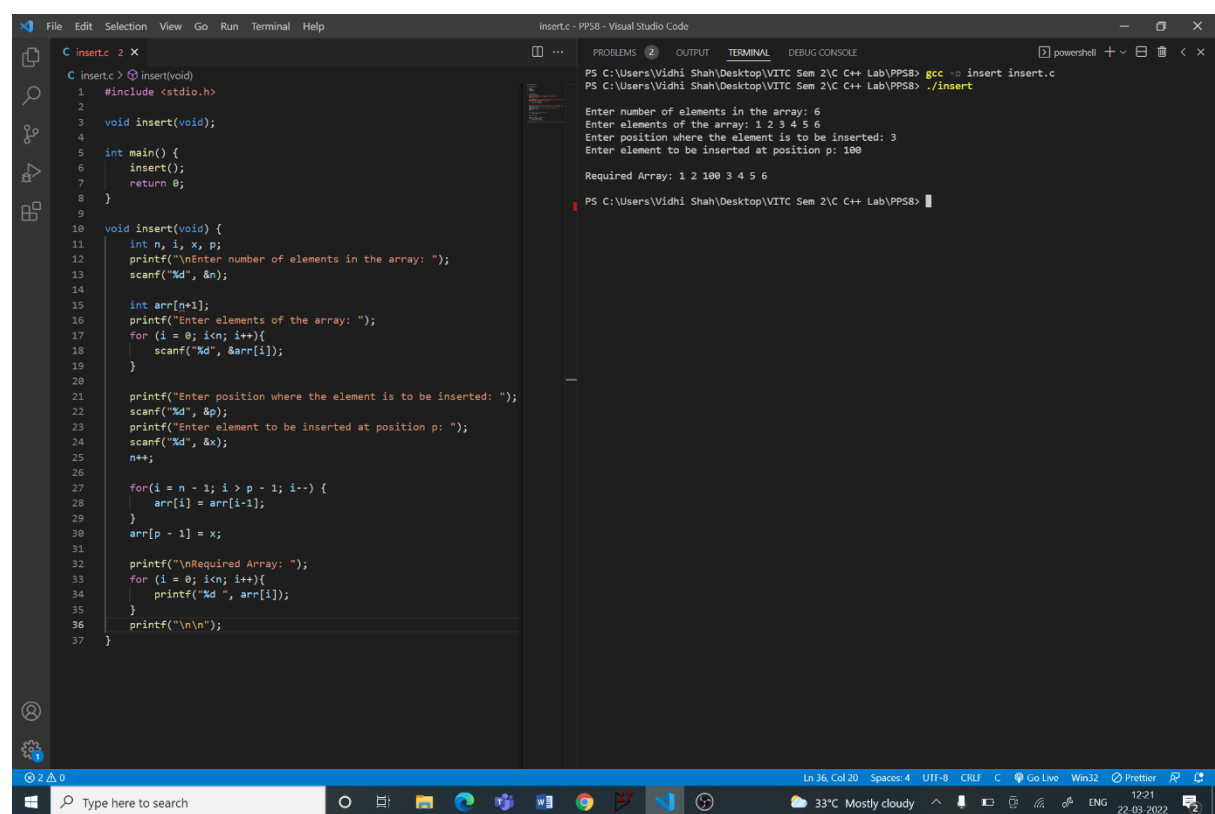
Step 7: For 'i' from n - 1 to p - 1 (excluding p - 1)

Step A: arr[i] = arr[i-1]

Step 8: arr[p - 1] = x

Step 9: Display the array

Code:



```
1 #include <stdio.h>
2
3 void insert(void);
4
5 int main() {
6     insert();
7     return 0;
8 }
9
10 void insert(void) {
11     int n, i, x, p;
12     printf("\nEnter number of elements in the array: ");
13     scanf("%d", &n);
14
15     int arr[n+1];
16     printf("Enter elements of the array: ");
17     for (i = 0; i < n; i++){
18         scanf("%d", &arr[i]);
19     }
20
21     printf("Enter position where the element is to be inserted: ");
22     scanf("%d", &p);
23     printf("Enter element to be inserted at position p: ");
24     scanf("%d", &x);
25     n++;
26
27     for(i = n - 1; i > p - 1; i--) {
28         arr[i] = arr[i-1];
29     }
30     arr[p - 1] = x;
31
32     printf("\nRequired Array: ");
33     for (i = 0; i < n; i++){
34         printf("%d ", arr[i]);
35     }
36     printf("\n\n");
37 }
```

PS C:\Users\Widhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> gcc -o insert insert.c

PS C:\Users\Widhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8> ./insert

Enter number of elements in the array: 6

Enter elements of the array: 1 2 3 4 5 6

Enter position where the element is to be inserted: 3

Enter element to be inserted at position p: 100

Required Array: 1 2 100 3 4 5 6

PS C:\Users\Widhi Shah\Desktop\WITC Sem 2\C++ Lab\PPS8>