Advanced Data Structures and Algorithms Laboratory (AI-525)

Submitted To:

Name: Dr. Simranjit Singh

Designation: Assistant Professor

Department: Information Technology

Submitted By:

Name: Rohit Kumar

Roll No: 25901334

Course: Mtech

Branch: AI

Sr.No	Experiment name	Date	Page No	Signature

LAB-1

Basic C Programs

Q1. Print "Hello, World!"

```
#include <stdio.h>
int main() {
	// your code goes here
	printf("Hello World");
}
```

Your Output

```
Hello World
```

Q2. Swap Two Numbers

```
#include <stdio.h>
int main() {
	// your code goes here
	int x,y;
	scanf("%d %d",&x,&y);
	printf("Enter two number: x : %d y : %d\n",x ,y);
	int temp=x;
	x=y;
	y=temp;
	printf("After swapping \n");
	printf("x = %d\n",x);
	printf("y = %d\n",y);
}
```

Sample Input

```
34 20
```

Your Output

```
Enter two number: x : 34 y : 20
After swapping
x = 20
y = 34
```

Q3. Check Even or Odd

```
#include <stdio.h>

int main() {
        // your code goes here
        int x;
        scanf("%d",&x);
        printf("Enter the number to check even or odd: x : %d\n",x);
        if(x%2==0){
           printf("The number %d is even",x);
   }
   else{
      printf("The number %d is Odd",x);
   }
   return 0;
}
```

Sample Input

```
20
```

Your Output

```
Enter the number to check even or odd: x : 20
The number 20 is even
```

Q4. Find Largest of Three Numbers

```
#include <stdio.h>

int main() {
        // your code goes here
        int x,y,z;
        scanf("%d%d%d",&x,&y,&z);
        printf("Enter three number to check which is larger: x : %d %d
%d\n",x,y,z);
        if(x==y && x==z){
           printf("The number %d is equal as other two",x);
   }
        else if(x>=y && x>=z){
           printf("The number %d is larger than other two",x);
   }
   else if(y>=x && y>=z){
      printf("The number %d is larger than other two",x);
   }
   else if(z>=x && z>=y){
      printf("The number %d is larger than other two",x);
   }
   return 0;
}
```

Sample Input

```
34 20 12
```

Your Output

```
Enter three number to check which is larger: x : 34 20 12
The number 34 is larger than other two
```

Q5. Simple Calculator (switch case)

```
#include <stdio.h>
int main() {

   double num1, num2;
   char op;
   scanf("%lf %lf %c", &num1, &num2, &op);
   printf("Enter two numbers and an operator (+, -, *, /): %lf %lf
%c\n",num1,num2,op);


   switch(op) {
      case '+':
         printf("Result = %.2lf\n", num1 + num2);
         break;
      case '-':
         printf("Result = %.2lf\n", num1 - num2);
         break;
      case '*':
         printf("Result = %.2lf\n", num1 * num2);
         break;
      case '/':
         if(num2 != 0)
            printf("Result = %.2lf\n", num1 / num2);
         else
            printf("Error! Division by zero not allowed.\n");
         break;
      default:
         printf("Invalid operator!\n");
   }
   return 0;
}
```

Sample Input

```
34 20 *
```

Your Output

```
Enter two numbers and an operator (+, -, *, /): 34.000000 20.000000 *
Result = 680.00
```

Q6. Factorial of a Number

```
#include <stdio.h>

long long fact(long long x) {
   if (x == 0 || x == 1) {
      return 1;
   }
   return x * fact(x - 1);
}

int main() {
   long long x;
   scanf("%lld", &x);
   printf("Enter number for factorial: %d\n",x);

   long long fac = fact(x);

   printf("The factorial of %lld is = %lld\n", x, fac);

   return 0;
}
```

Sample Input

```
34
```

Your Output

```
Enter number for factorial: 34
The factorial of 34 is = 4926277576697053184
```

Q7. Fibonacci Series (first n terms)

```
#include <stdio.h>

int main() {
   int n, i;
   long long t1 = 0, t2 = 1, nextTerm;

   scanf("%d", &n);
   printf("Enter till number you want Fibonacci : %d\n",n);

   printf("Fibonacci Series: ");
   for (i = 1; i <= n; i++) {
      printf("%lld ", t1);
      nextTerm = t1 + t2;
      t1 = t2;
      t2 = nextTerm;
   }
   printf("\n");

   return 0;
}
```

Sample Input

```
34
```

Your Output

```
Enter till number you want Fibonacci : 34
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711
28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578
```

Q8. Reverse a Number

```
#include <stdio.h>

int main() {
   int n;
   scanf("%d", &n);
   printf("Enter number you want to reverse : %d\n",n);

   int original=n;
   int rem=0;
   int reversed=0;
   while (n != 0) {
      rem = n % 10;
      reversed = reversed * 10 + rem;
      n /= 10;
   }
   printf("Reversed number of %d is: %d\n", original, reversed);

   return 0;
}
```

Sample Input

```
34
```

Your Output

```
Enter number you want to reverse : 34
Reversed number of 34 is: 43
```

Q9. Palindrome Number Check

```
#include <stdio.h>

int main() {
   int n;
   scanf("%d", &n);
   printf("ATM checks if PIN entered forward = reverse give pin: %d\n",n);

   int original=n;
   int rem=0;
   int reversed=0;
   while (n != 0) {
      rem = n % 10;
      reversed = reversed * 10 + rem;
      n /= 10;
   }
   if(original==reversed){
      printf("PIN you entered is pallindrome of %d is %d\n", original,reversed);
   }
   else{
      printf("PIN you entered is not pallindrome as %d\n", original);
   }

   return 0;
}
```

Sample Input

```
3420
```

Your Output

```
ATM checks if PIN entered forward = reverse give pin: 3420

PIN you entered is not pallindrome as 3420
```

Q10. Count Digits in a Number

```
#include <stdio.h>

int main() {
   int n;
   scanf("%d", &n);
   printf("Give number i will count how many digit in number: %d\n",n);

   int original=n;
   int count=0;
   if(n==0){
      count=1;
   }
   else{
      while(n>0){
         n/=10;
         count++;
      }
   }
   printf("The digit in number %d is %d\n",original,count);

   return 0;
}
```

Sample Input

```
3420
```

Your Output

```
Give number i will count how many digit in number: 3420
The digit in number 3420 is 4
```

Q11. Sum of Digits

```
#include <stdio.h>

int main() {
   long long n;
   scanf("%lld", &n);
   printf("Give number i will sum of all digit in number you gave: %lld\n",n);

   long long original=n;
   long long sum=0;
   long long count=0;

   while(n!=0){
      sum+=(n%10);
      n/=10;
   }
   printf("The Sum digit in number %lld is %lld\n",original,sum);

   return 0;
}
```

Sample Input

```
2034
```

Your Output

```
Give number i will sum of all digit in number you gave: 2034
The Sum digit in number 2034 is 9
```

Q12. Check Prime Number

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

bool isit(long long n){
   if(n==0 || n==1){
      return 0;
   }
   if(n==2){
      return 1;
   }
   for(int i=3;i*i<=n;i+=2){
      if(n%i){
         return 0;
      }
   }
   return 1;
}

int main() {
   long long n;
   scanf("%lld", &n);
   printf("Give number lets check it is prime or not : %lld\n",n);

   bool hmm=isit(n);
   if(!hmm){
      printf("The number %lld you gave is not prime\n",n);
   }
   else{
      printf("The number %lld you gave is prime\n",n);
   }
   return 0;
}
```

Sample Input

```
2034
```

Your Output

```
Give number lets check it is prime or not : 2034
The number 2034 you gave is not prime
```

Q13. Array – Find Maximum Element

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

int main() {
   long long n;
   scanf("%lld", &n);
   printf("Give array size later input array elements, lets find Largest number
you gave in array : %lld\n",n);
   long long ths[n];
   long long maxi=LLONG_MIN;
   for(int i=0;i<n;i++){
      scanf("%lld",&ths[i]);
      if(ths[i]>maxi) {
         maxi=ths[i];
      }
   }

   printf("The Largest number in your given array is %lld\n",maxi);

   return 0;
}
```

Sample Input

```
3
34 20 12
```

Your Output

```
Give array size later input array elements, lets find Largest number you gave in array : 3
The Largest number in your given array is 34
```

Q14. String – Count Vowels

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

int main() {
   char *str = NULL;
   size_t size = 0;
   int count = 0;

   printf("Enter a string, lets check how many vowel in it\n");
   ssize_t len = getline(&str, &size, stdin);

   if (len != -1) {
      for (ssize_t i = 0; i < len; i++) {
         char ch = tolower(str[i]);
         if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
            count++;
         }
      }
   }
   printf("The given string has %d vowels in it\n", count);
   return 0;
}
```

Sample Input

```
Ohi
```

Your Output

```
Enter a string, lets check how many vowel in it

The given string has 2 vowels in it
```

Q15. Scenario – Electricity Bill Calculation

```
#include <stdio.h>

int main() {
   int units;
   int bill = 0;

   printf("Enter the number of electricity units consumed: \n");
   scanf("%d", &units);

   if (units <= 100) {
      bill = units*5;
   }
   else if (units <= 200) {
      bill = 100*5+(units-100)*7;
   }
   else {
      bill = 100*5+100*7+(units-200)*10;
   }

   printf("Total electricity bill of %d units is : Rs %d\n", units,bill);

   return 0;
}
```

Sample Input

```
2034
```

Your Output

```
Enter the number of electricity units consumed:

Total electricity bill of 2034 units is : Rs 19540
```

Q16. Factorial using Recursion

```
#include <stdio.h>
#define long long int

int fac(int n){
   if(n==0 || n==1){
      return 1;
   }
   return n*fac(n-1);
}

int main() {
   int x;
   printf("Enter a number to calculate Factorial \n");
   scanf("%d", &x);
   if(x<0){
      printf("The Factorial of number is not possible as you gave negative
number : %d\n", x);
      return 0;
   }
   int fact=fac(x);
   printf("The Factorial of number %d is : %d\n", x,fact);

   return 0;
}
```

Sample Input

```
6
```

Your Output

```
Enter a number to calculate Factorial
The Factorial of number 6 is : 720
```

Q17. Fibonacci Series using Recursion

```
#include <stdio.h>
#define long long int

int fib(int n){
   if(n==0){
      return 0;
   }
   if(n==1){
      return 1;
   }
   return fib(n-1)+fib(n-2);
}

int main() {
   int n;
   printf("Enter a number to calculate Fibonacci \n");
   scanf("%d", &n);
   if(n<0){
      printf("The Fibonacci of number is not possible as you gave negative
number : %d\n", n);
      return 0;
   }

   printf("The Fibonacci of number is : \n");
   for(int i=0;i<n;i++){
      printf("%d ",fib(i));
   }

   return 0;
}
```

Sample Input

```
6
```

Your Output

```
Enter a number to calculate Fibonacci
The Fibonacci of number is :
0 1 1 2 3 5
```

Q18. GCD (Greatest Common Divisor) using Recursion

```
#include <stdio.h>

int gcd(int n1,int n2){
   if(n2==0){
      return n1;
   }
   return gcd(n2, n1%n2);
}

int main() {
   int n1,n2;
   printf("Enter two number to calculate GCD \n");
   scanf("%d%d", &n1,&n2);
   if(n1==0 && n2==0){
      printf("The GCD of number is not possible\n");
      return 0;
   }

   printf("The GCD of number is : ");
   int x=gcd(n1,n2);
   printf("%d",x);

   return 0;
}
```

Sample Input

```
34 20
```

Your Output

```
Enter two number to calculate GCD
The GCD of number is : 2
```

Q19. Sum of Digits using Recursion

```
#include <stdio.h>

int sumOfDigits(int n) {
   if (n == 0)
      return 0;
   return (n % 10) + sumOfDigits(n / 10);
}

int main() {
   int num;
   printf("Enter a number to find the sum of its digits: \n");
   scanf("%d", &num);

   if (num < 0) {
      num = -num;
   }

   int sum = sumOfDigits(num);
   printf("The sum of digits of %d is: %d\n", num, sum);

   return 0;
}
```

Sample Input

```
34
```

Your Output

```
Enter a number to find the sum of its digits:
The sum of digits of 34 is: 7
```

Q20. Recursive Binary Search

```
#include <stdio.h>

int binarySearch(int arr[], int left, int right, int key) {
   if(left > right){
      return -1;
   }
   else{
      int mid = left + (right - left) / 2;

      if(arr[mid] == key){
         return mid;
      }
      else if(arr[mid] < key){
         return binarySearch(arr, mid + 1, right, key);
      }
      else{
         return binarySearch(arr, left, mid - 1, key);
      }
   }
}

int main() {
   int n, key;


   scanf("%d", &n);
   printf("Enter the number of book IDs: %d\n",n);

   int arr[n];
   printf("Enter the book IDs: \n");
   for(int i = 0; i < n; i++){
      scanf("%d", &arr[i]);
      printf("%d ",arr[i]);
   }
   printf("\n");


   scanf("%d", &key);
   printf("Enter the book ID to search: %d\n",key);
```

```
  int result = binarySearch(arr, 0, n - 1, key);

  if(result != -1){
    printf("Book ID %d found at index %d\n", key, result);
  }
  else{
    printf("Book ID %d not found in the system\n", key);
  }

  return 0;
}
```

Sample Input

```
5
34 2 20 12 6
20
```

Your Output

```
Enter the number of book IDs: 5
Enter the book IDs:
34 2 20 12 6
Enter the book ID to search: 20
Book ID 20 found at index 2
```

LAB-2

Time and Space Complexity Analysis of Fundamental C Programming Problems

1.A.Write a simple iterative program to compute the sum of elements.

```
#include <stdio.h>

int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        long long sum=0;
        for(int i=0;i<n;i++){

           int x;
           scanf("%d",&x);
           sum+=x;
        }
        printf("%ld",sum);
        return 0;
}
```

Sample Input

```
6
20 34 6 12 20 24
```

Your Output

```
116
```

1.B. Write a recursive version of the same program.

```
#include <stdio.h>

int sum(int arr[],int n){
   if(n==0){
      return 0;
   }
   return arr[n-1]+sum(arr,n-1);
}

int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        int arr[n];
        for(int i=0;i<n;i++){
           int x;
           scanf("%d",&arr[i]);
        }
        long long ans=sum(arr,n);
        printf("%ld",ans);
        return 0;
}
```

Sample Input

```
6
20 34 6 12 20 24
```

Your Output

```
116
```

2.A. Find maximum element using a simple linear scan (O(n)).

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

int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        int maxi=INT_MIN;
        for(int i=0;i<n;i++){
           int x;
           scanf("%d",&x);
           if(x>maxi){
              maxi=x;
           }
        }
        printf("%ld",maxi);
        return 0;
}
```

Sample Input

```
6
20 34 6 12 20 24
```

Your Output

```
34
```

2.B. Find maximum element Can you find it by first sorting the array and taking the last element

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

void bubbleSort(int arr[], int n) {
   for(int i =0;i<n-1;i++){
      for(int j =0;j<n-i-1;j++){
         if (arr[j] > arr[j + 1]){
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }
}


int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        int arr[n];
        for(int i=0;i<n;i++){
           int x;
           scanf("%d",&arr[i]);
        }
        printf("Maximum Element\n");
        bubbleSort(arr,n);
        printf("%d ",arr[n-1]);
        return 0;
}
```

Sample Input

```
6
20 34 6 12 20 24
```

Your Output

```
34
```

3.A. Reverse in-place using two-pointer technique (O(n), O(1)).

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

int main() {
        // your code goes here
        char str[1000];
        scanf("%s",str);
        int n=strlen(str);
        int start=0;
        int end=n-1;
        while(start<end){
           char temp=str[end];
           str[end]=str[start];
           str[start]=temp;
           start--;
           end++;
        }
        printf("%s",str);
        return 0;
}
```

Sample Input

taibhcodreafsgrhaiv

Your Output

vaibhcodreafsgrhait

3.B.Reverse by creating a new array (O(n), O(n)).

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

int main() {
	// your code goes here
	char str[1000];
	scanf("%s",str);
	int n=strlen(str);
	int end=n-1;
	char ans[1000];

	for(int i=end;i>=0;i--){
	   ans[end-i]=str[i];
	}
	printf("%s",ans);
	return 0;
}
```

Sample Input

taibhcodreafsgrhaiv

Your Output

vaibhcodreafsgrhait

4.A.Using modulo operator (O(1)).

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


int main() {
	// your code goes here
	int n;
	scanf("%d",&n);
	if(n%2==0){
	   printf("It is Even Number");
	}
	else{
	   printf("It is ODD Number");
	}
	return 0;
}
```

Sample Input

34

Your Output

It is Even Number

4.B. Using bitwise AND (n & 1) (O (1)).

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


int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        if((n&1)==0){
           printf("It is Even Number");
        }
        else{
           printf("It is ODD Number");
        }
        return 0;
}
```

Sample Input

```
34
```

Your Output

```
It is Even Number
```

5.A.Iterative method (O(n), O(1)).

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


int main() {
	// your code goes here
	int n;
	scanf("%d",&n);
	long long ans=1;
	for(int i=1;i<=n;i++){
	   ans*=i;
	}
	printf("%lld ",ans);
	return 0;
}
```

Sample Input

34

Your Output

4926277576697053184

5.B. Recursive method (O(n), O(n) for call

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

int fact(int n){
   if(n==0 || n==1){
      return 1;
   }
   return n*fact(n-1);
}


int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        int ans=fact(n);
        printf("%d ",ans);
        return 0;
}
```

Sample Input

34

Your Output

4926277576697053184

6.A. Implement basic linear search (O(n), O(1)).

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


int main() {
	// your code goes here
	int n;
	scanf("%d",&n);
	int arr[n];
	for(int i=0;i<n;i++){
	   int x;
	   scanf("%d",&arr[i]);
	}
	int target;
	bool isit=false;
	scanf("%d",&target);

	for(int i=0;i<n;i++){
	   if(arr[i]==target){
	      isit=true;
	      break;
	   }
	}
	if(isit){
	   printf("The target value is in array");
	}
	else{
	   printf("The target value is NOT in array");
	}
	return 0;
}
```

Sample Input

```
5
34 20 6 12 24
12
```

Your Output

```
The target value is in array
```

6.B.Optimize by using sentinel method to reduce comparisons (still O(n), but fewer

```
#include <stdio.h>
#include <stdbool.h>
int main() {
	// your code goes here
	int n;
	scanf("%d",&n);
	int arr[n];
	for(int i=0;i<n;i++){
	  int x;
	  scanf("%d",&arr[i]);
	}
	int target;
	scanf("%d",&target);
	int start=0;
	int end=n-1;
	bool isit=false;
	while(start<=end){
	  if(arr[start]!=target){
	    isit=true;
	    start++;
	  }
	  else if(arr[end]!=target){
	    isit=true;
	    end--;
	  }
	  else{
	    start++;
	    end--;
	  }
	}
	if(isit){
	  printf("The target value is in array");
	}
	else{
	  printf("The target value is NOT in array");
	}
	return 0;
}
```

Sample Input

```
5
34 20 6 12 24
12
```

Your Output

```
The target value is in array
```

7.A.Print First n Natural Numbers using simple loop

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

int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        for(int i=1;i<=n;i++){
           printf("%d ",i);
        }
        return 0;
}
```

Sample Input

```
20
```

Your Output

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

7.B.Print First n Natural Numbers Using recursion (O(n), O(n)).

```
#include <stdio.h>
#include <stdbool.h>
void natural(int n,int i){
   if(i>n){
      return;
   }
   printf("%d ",i);
   return natural(n,i+1);
}


int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        natural(n,1);
        return 0;
}
```

Sample Input

```
20
```

Your Output

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

8.A.Count Vowels in a String Traverse string and check each character (O(n), O(1)).

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

int main() {
        // your code goes here
        char str[1000];
        scanf("%s",str);
        int count=0;
        int n=strlen(str);
        for(int i=0;i<n;i++){
           char ch=tolower(str[i]);
          // ch=ch.tolower(ch);
           if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){
              count+=1;
           }
        }
        printf("The vowels in string is %d ",count);
        return 0;
}
```

Sample Input

```
vaibhcodreafsgrhait
```

Your Output

```
The vowels in string is 7
```

8.B.Count Vowels in a String Use a lookup table (array of 256 size) to speed up vowel checking (O(n), O(1) but extra

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

int main() {
        // your code goes here
        char str[1000];
        scanf("%s",str);
        int count=0;
        int n=strlen(str);
        bool isVowel[256] = {false};
   isVowel['a'] = true;
   isVowel['e'] = true;
   isVowel['i'] = true;
   isVowel['o'] = true;
   isVowel['u'] = true;
   isVowel['A'] = true;
   isVowel['E'] = true;
   isVowel['I'] = true;
   isVowel['O'] = true;
   isVowel['U'] = true;
        for(int i=0;i<n;i++){
           if(isVowel[str[i]]){
              count+=1;
           }
        }
        printf("The vowels in string is %d ",count);
        return 0;
}
```

Sample Input

```
vaibhcodreafsgrhait
```

Your Output

```
The vowels in string is 7
```

9.A.Swap Two Numbers Without Temporary Variable Using arithmetic (a = a + b; b = a - b; a = a - b).

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

int main() {
	// your code goes here
	int n,m;
	scanf("%d%d",&n,&m);
	n=n+m;
	m=n-m;
	n=n-m;
	printf("The Swaps of number is %d and %d",n,m);
	return 0;
}
```

Sample Input

```
20 34
```

Your Output

```
The Swaps of number is 34 and 20
```

9.B.Swap Two Numbers Without Temporary Variable Using bitwise XOR (a = a ^ b; b = a ^ b; a = a ^ b).

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

int main() {
        // your code goes here
        int n,m;
        scanf("%d%d",&n,&m);
        n=n^m;
        m=n^m;
        n=n^m;
        printf("The Swaps of number is %d and %d",n,m);
        return 0;
}
```

Sample Input

```
20 34
```

Your Output

```
The Swaps of number is 34 and 20
```

4210.A.Check Palindrome Number Convert number to string and check (O(n), O(n)).

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

int main() {
        // your code goes here
        int n;
        scanf("%d",&n);
        char str[20];
        sprintf(str, "%d", n);
        bool isit=true;
        int low=0;
        int high=strlen(str)-1;
        while(low<high){
          if(str[low]!=str[high]){
            isit=false;
            break;
          }
          low++;
          high--;
        }
        if(isit){
          printf("Yes the number is Palindrome Number");
        }
        else{
          printf("No the number is Palindrome Number");
        }
        return 0;
}
```

Sample Input

34643

Your Output

Yes the number is Palindrome Number

10.B.Check Palindrome Number Reverse digits without converting (O (log n), O (1)).

```
#include <stdio.h>

int main() {
   int n, original, rev = 0;
   scanf("%d", &n);
   original = n;

   while (n > 0) {
      int digit = n % 10;
      rev = rev * 10 + digit;
      n /= 10;
   }

   if (rev == original){
            printf("Yes the number is Palindrome Number");
         }
         else{
            printf("No the number is Palindrome Number");
         }

   return 0;
}
```

Sample Input

34643

Your Output

Yes the number is Palindrome Number