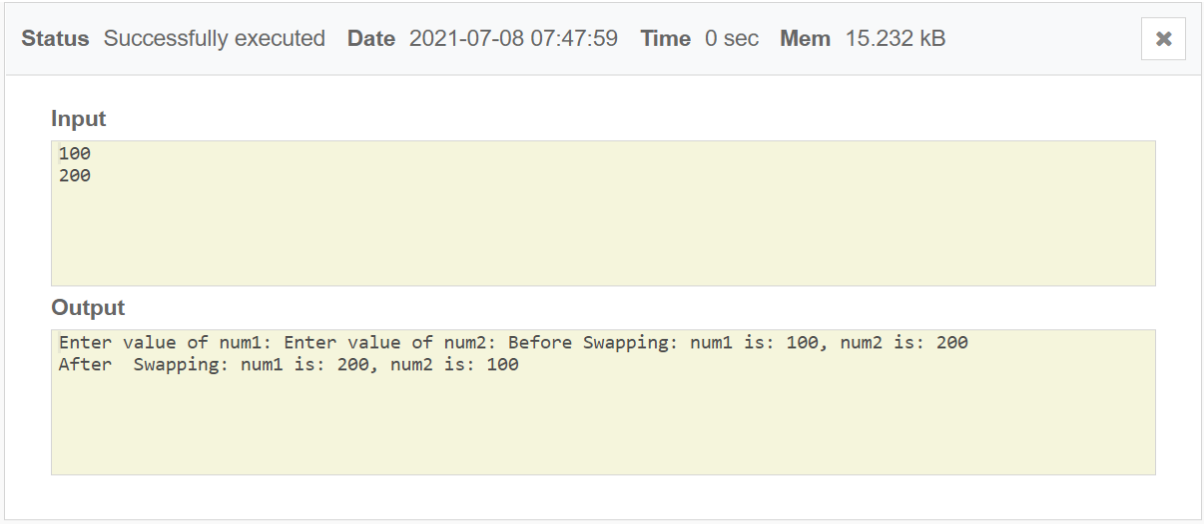
CN LAB 1

Name-Ritam Hudait

Roll-1905797

1. Write a C program to swap the content of 2 variables using pointer.

```
#include <stdio.h>
void swap(int *a,int *b)
{
   int temp;
    temp   = *a;
   *a   = *b;
   *b   =  temp;
}
int main()
{
   int num1,num2;
   printf("Enter value of num1: ");
   scanf("%d",&num1);
   printf("Enter value of num2: ");
   scanf("%d",&num2);
   printf("Before Swapping: num1 is: %d, num2 is: %d\n",num1,num2);
   swap(&num1,&num2);
   printf("After  Swapping: num1 is: %d, num2 is: %d\n",num1,num2);
   return 0;
}
```

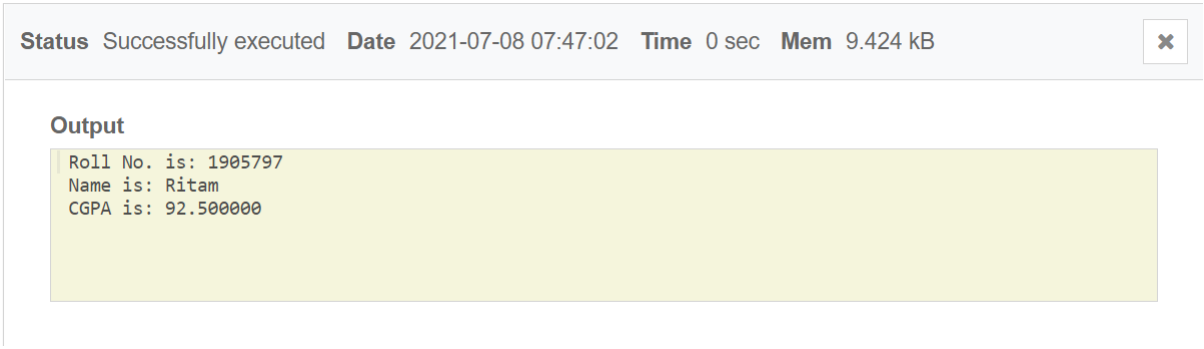


2. Write a C program to assign values to each members of the following structure. Pass the populated structure to a function Using call-by address and print the value of each member of the structure with in that function.

```
struct info{
int roll_no;
char name[50];
float CGPA;
}
#include <stdio.h>
#include <string.h>
struct info
{
        int roll_no;
        char name[50];
        float CGPA;
};
void func(struct info *student);
int main()
{
        struct info student;
        student.roll_no=1905797;
```

```
        strcpy(student.name, "Ritam");
        student.CGPA = 92.5;
        func(&student);
        return 0;
}

void func(struct info *student)
{
        printf(" Roll No. is: %d \n", student->roll_no);
        printf(" Name is: %s \n", student->name);
        printf(" CGPA is: %f \n", student->CGPA);
}
```

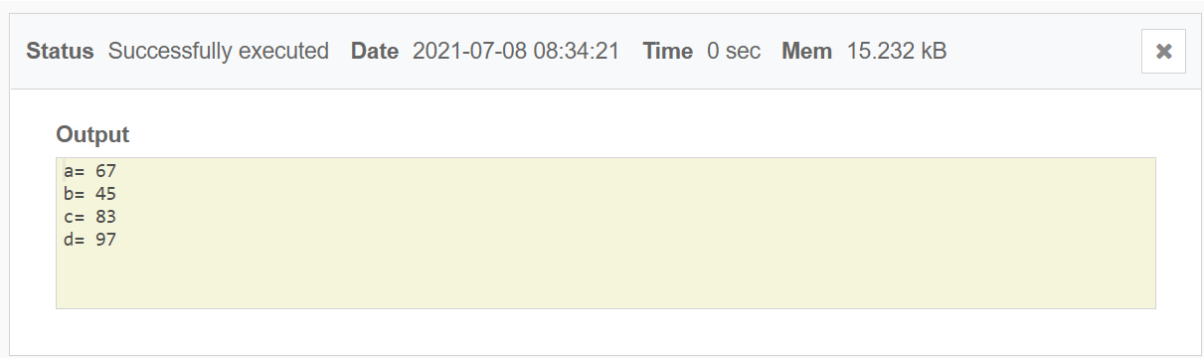


3. Write a C program to extract each byte from a given number and store them in separate character variables and print the content of those variables.

```
  #include <stdio.h>
int main()
{
   unsigned int x=0x97834567;
   int a,b,c,d;
   a=(x&0xFF);
   b=((x>>8)&0xFF);
   c=((x>>16)&0xFF);
   d=((x>>24)&0xFF);

   printf("a= %02X\n",a);
```

```
    printf("b= %02X\n",b);
    printf("c= %02X\n",c);
    printf("d= %02X\n",d);
    return 0;
}
```

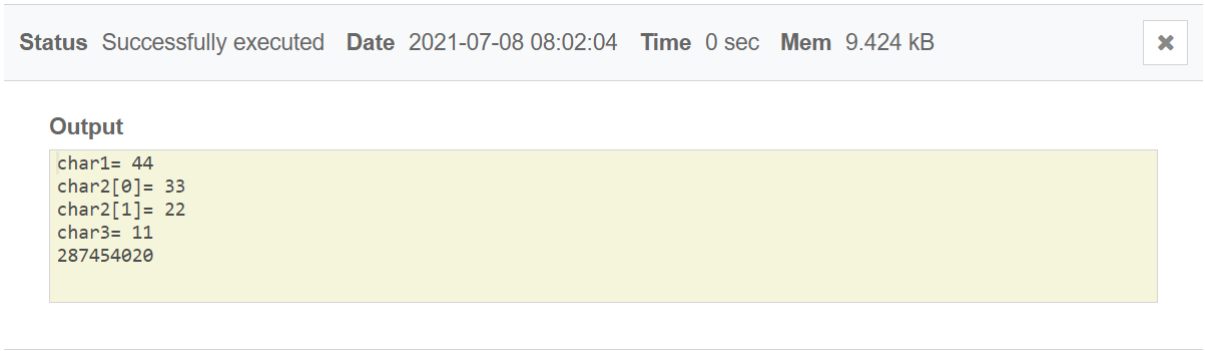


4. Write a C Program to enter a number and store the number across the following structure and print the content of each member of the structure. Then aggregate each member of the structure to form the original number and print the same.

```
struct pkt{
char ch1;
char ch2[2];
char ch3;
};
#include<stdio.h>
struct pkt{
    unsigned char ch1;
unsigned char ch2[2];
    unsigned char ch3;
};
int main(){
    unsigned int value=0x11223344;
    int a,b,c,d;
     a=(value&0xFF);
    b=((value>>8)&0xFF);
```

```
    c=((value>>16)&0xFF);
    d=((value>>24)&0xFF);
    struct pkt ob1;
    ob1.ch1=a;
    ob1.ch2[0]=b;
    ob1.ch2[1]=c;
    ob1.ch3=d;

    printf("char1= %.2x",ob1.ch1);
    printf("\nchar2[0]= %.2x",ob1.ch2[0]);
    printf("\nchar2[1]= %.2x",ob1.ch2[1]);
    printf("\nchar3= %.2x",ob1.ch3);
    int val=((int)(ob1.ch3<<24)+(int)(ob1.ch2[1]<<16)+(int)(ob1.ch2[0]<<8)+(int)(ob1.ch1));
    printf("\n%d",val);
    return 0;
}
```



5. Write a C program to check whether the Host machine is in Little Endian or Big Endian. Enter a number, print the content of each byte location and Convert the Endianness of the same i.e. Little to Big Endian and vice-versa.

```
#include<stdio.h>

void printing(unsigned char *val,int n){
    int k;
```

```
    for(k=0;k<n;k++){
        printf("%.2x  ",val[k]);
    }
    if(val[0]=44){
        printf("Little Endian");
    }
    else
        printf("Big Endian");

    printf("\n");
    printf("Little Endian::");
    for(k=0;k<n;k++){
        printf("%.2x  ",val[k]);
    }
    printf("Big Endian::");
    for(k=n-1;k>-1;k--){
        printf("%.2x  ",val[k]);
    }
}

int main()
{
    int i=1234567;
    printing((unsigned char*)&i,sizeof(i));
    return(0);
}
```

Status Successfully executed **Date** 2021-07-08 08:11:16 **Time** 0 sec **Mem** 15.232 kB

Output

```
87  d6  12  00  Little Endian
Little Endian::2c  d6  12  00  Big Endian::00  12  d6  2c
```