#include <stdio.h>

#include <stdlib.h>

int main()

{

    int num = 65;

    printf("%d\n", num);

    printf("%4d\n", num);

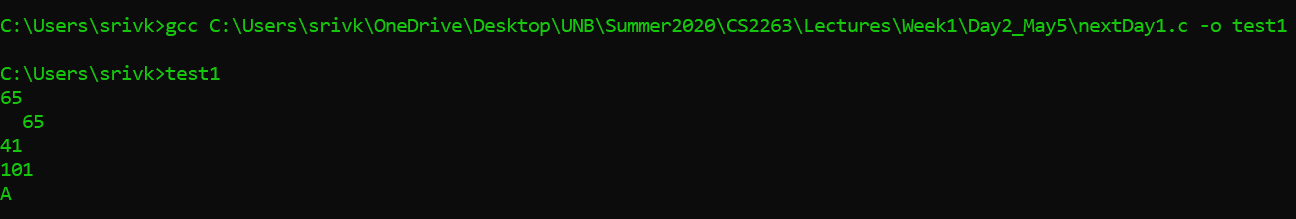
    printf("%x\n", num);

    printf("%o\n", num);

    printf("%c\n", (char)num);

}

Output:



#include <stdio.h>

#include <stdlib.h>

int main()

{

    char ch = 'b';

    int num = 11;

    float num1 = 14.567;

    double num2 = 10.0;

    printf("The size of the character is %d\n", sizeof(ch));

    printf("The size of the int is %d\n", sizeof(num));

    printf("The size of the float is %d\n", sizeof(num1));

    printf("The size of the double is %d\n", sizeof(num2));

}

Output:

A close up of a logo

Description automatically generated

These numbers are the amount of memory allocated to that datatype.

<limit.h>

#include <stdio.h>

#include <limits.h>

int main()

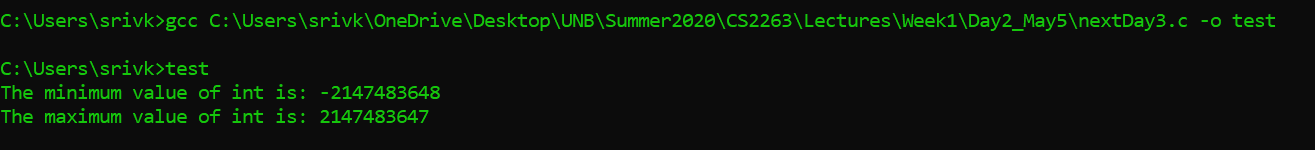
{

    printf("The minimum value of int is: %d\n", INT\_MIN);

    printf("The maximum value of int is: %d\n", INT\_MAX);

}

Output:



**playStack.c**

// first.c

#include <stdio.h>

#include <stdlib.h>

#define MAX 256

#define PUSH 1

#define POP 0

#define LIST 2

int main(int argc, char\* argv[])

{

    int count = 0;

    int i;

    int stack[MAX];

    int size = 0;

    int val;

    int iChoice;

    int iNRead;

  /\* Processing loop \*/

  printf("Choice (1=add, 0=remove, 2=list): ");

  iNRead = scanf("%d", &iChoice);

  while(iNRead == 1)

  {

    switch(iChoice)

    {

      case PUSH:

        printf("Value to add: ");

        // Read the element, add it to the stack

        int valE;

        valE = scanf("%d", &val);

        if(valE != 1)

        {

          printf("Invalid input\n");

        }

        if(size < MAX)

        {

            stack[size] = val;

            size++;

        }

        else

        {

            printf("Error. Exceeding the MAX size");

        }

      break;

      case POP:

      // Print out the last element and remove it.

        if(size > 0)

        {

            val = stack[size - 1];

            printf("Removed element: %d\n", val);

            size--;

        }

        else

        {

          printf("Stack is empty");

        }

      break;

      case LIST:

        // Print out the stack elements

            for(i = 0; i < size; i++)

            {

                printf("\n%d", stack[i]);

                printf("\n");

            }

      break;

    }

    printf("Choice (1=add, 0=remove, 2=list): ");

    iNRead = scanf("%d", &iChoice);

  }

  return EXIT\_SUCCESS;

}

Output:

