1.

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \* \* argv)

{

int a = 65;

printf("%d, %d, %4d, %x, %o, %c",65,65,65,65,65,(char)65);

}

文本

描述已自动生成

**2.**

#include <stdio.h>

int main(int argc, char \* \* argv){

int a;

char b;

float c;

double d;

printf("size of int %lu\n", sizeof(a));

printf("size of char %lu\n", sizeof(b));

printf("size of float %lu\n", sizeof(c));

printf("size of double %lu\n", sizeof(d));

}

文本

描述已自动生成

The size here is the number of bytes for each datatype.

**3.**

/\* Minimum and maximum values a `signed int' can hold. \*/

# define INT\_MIN (-INT\_MAX - 1)

# define INT\_MAX 2147483647

So -2147483648 to 2147483647

**4.**

// 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 stack[MAX];

int size = 0;

int val;

int iChoice;

int iNRead;

int result;

int newElement;

int read;

/\* 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: ");

iNRead = scanf("%d", &newElement);

if(size < MAX){

stack[size] = newElement;

size++;

}

// Read the element, add it to the stack

break;

case POP:

result = stack[size-1];

stack[size] = 0;

size--;

printf("Value poped: %d", result);

// Print out the last element and remove it.

break;

case LIST:

for(int i=0;i<size;i++){

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

}

break;

}

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

iNRead = scanf("%d", &iChoice);

}

return EXIT\_SUCCESS;

}

文本

描述已自动生成