#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 100 // Maximum size of the stack

// Part (a): typedef for representing a stack of integers using an array

typedef struct {

int data[MAX\_SIZE];

int top; // Index of the top element in the stack

} Stack;

// Part (b): Function to initialize the stack

Stack init() {

Stack stack;

stack.top = -1; // Stack is initially empty

return stack;

}

// Part (b): Function to check if the stack is empty

int empty(Stack \*stack) {

return stack->top == -1;

}

// Part (b): Function to push an element onto the stack

void push(Stack \*stack, int value) {

if (stack->top == MAX\_SIZE - 1) {

printf("Stack Overflow\n");

return;

}

stack->data[++stack->top] = value;

}

// Part (b): Function to return the top element of the stack

int top(Stack \*stack) {

if (empty(stack)) {

printf("Stack is empty\n");

exit(1);

}

return stack->data[stack->top];

}

// Part (b): Function to delete the top element from the stack

void delete(Stack \*stack) {

if (empty(stack)) {

printf("Stack is empty\n");

exit(1);

}

stack->top--;

}

// Part (c): Function to print an unsigned integer to base 5 using a stack

void printBase5(unsigned int num) {

Stack stack = init();

while (num > 0) {

int remainder = num % 5;

push(&stack, remainder);

num /= 5;

}

printf("Base 5 representation: ");

while (!empty(&stack)) {

printf("%d", top(&stack));

delete(&stack);

}

printf("\n");

}

int main() {

unsigned int number = 27; // Example number to print in base 5

printBase5(number);

return 0;

}