struct Stack {

массив и количество (или список)

};

void st\_init(Stack\*st) // инициализировать стек

void st\_pop(Stack\*st) // удаление вершины стека

void st\_push(Stack\*st, double x) // добавление в стек

double st\_top(Stack\*st) // вернуть вершину стека без удаления

void st\_replace(Stack\*st, double x) // заменить вершину стека

Команды хранятся в массиве структур cmd:

struct Command {

 char \*name;

 void (\*func)(Stack \*);

} cmd[7]=

{ {“clear”,do\_clear}, … } // инициализация массива

Где функция do\_clear:

void do\_clear(Stack \*st)

{ st\_init(st);

}

Выполнение i-й команды: (\*cmd[i].func)(&st);

Написать функцию main, в которой в цикле вводится команда или число, если введенное значение это число, то оно добавляется в стек, иначе поиск команды в массиве cmd и выполнение

int main()

{ Stack st;

 char s[100];

 st\_init(&st);

 while(scanf(“%s”,s)==1)

 { if(s это число)

 { double x;

 sscanf(s,”%lf”,&x);

 st\_push(&st,x);

 }

 else

 { for(i по командам)

 if(strcmp(cmd[i].name,s)==0)

 { (\*cmd[i].func)(&st);

 break;

 }

 }

 }

}

Обязательные команды + - \* / print clear quit

И команды в соответствии с вариантом

Пример работы

* 2.5 4 + print

6.5

* 2 \* print

13

* quit

Программа должна быть надежной, т.е. завершаться только при вводе quit, но сообщать о проблемах (деление на 0, переполнение стека и т.д.)

Написать отчет

Варианты

1. sin, cos, tan, atan, asin, acos, pi (константа)
2. exp, log, ^, sqrt, hypot, e (константа)
3. вместо double комплексные числа
4. вместо double int и % and or xor << >> oct dec (база системы счисления чисел)
5. вместо double long long и % & | ^ << >> hex dec (база системы счисления чисел)