

Описание задания:

Номер варианта: 298

Номер задачи: 4

Номер функции обработки данных в контейнере: 22

Обобщенный артефакт, используемый в задании	Базовые альтернативы	Общие переменные	Общая функция
4. Объемная (трехмерная) геометрическая фигура.	1. Шар (целочисленный радиус) 2. Параллелепипед (три целочисленных ребра) 3. Правильный тетраэдр (длина ребра – целое)	Плотность материала фигуры (действительное число)	Вычисление площади поверхности (действительное число)

Функция:

22.Переместить в конец контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, больше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы сдвинуть к началу без изменения их порядка.

Ниже приведена таблица типов:

Таблица типов			
Название	Размер, Байт	Название	Размер, Байт
base type: int	4	struct ball	16
base type: bool	1	radius: int	4
base type: double	8	density: double	8
base type: char**	8	struct parallelepiped	20
base type: FILE	216	edge_a: int	4
struct container	320004	edge_b: int	4
len: int	4	edge_c: int	4
cont: *shape[]	320000	density: double	8
struct shape	32	struct tetrahedron	16
k: enum key{}	4	edge: int	4
ball: ball	16	density: double	8
parallelepiped: parallelepiped	20		
tetrahedron: tetrahedron	16		

Ниже приведено описание работы функции main в рамках архитектуры:

Heap	Память программы	
argv[0] = "/bin/project"	исходный файл main.cpp	размер
argv[1] = "-f"	average: double	8
argv[2] = "iofiles/test1.txt"	time: unsigned int	4
argv[3] = "iofiles/out.txt"	argc: int	4
argv[4] = "iofiles/out2.txt"	argv: char**	8
	container: container	320004
Stack	p_file: FILE	216
ShowErrorMessage	is_file_input: bool	1
Init	size: int	4
InContainer	Глобальная память	
InRndContainer	-	
OutContainer		
RearrangeContainer		
OutContainer		
Clear		

Описание работы функции RearrangeContainer в рамках архитектуры:

Stack	Память программы	
RearrangeContainer	void RearrangeContainer(...)	
Square?		Память, Байт
memcpy	c: container	320004
	average: double	8
	size: int	4
Heap	cpy_array: shape*	320004
c: container	i: int	4
	j: int	4
Глобальная память		
-		

Примечание: вопросительный знак означает, что вызов функции зависит от условий.

Описание работы функции AverageMean в рамках архитектуры:

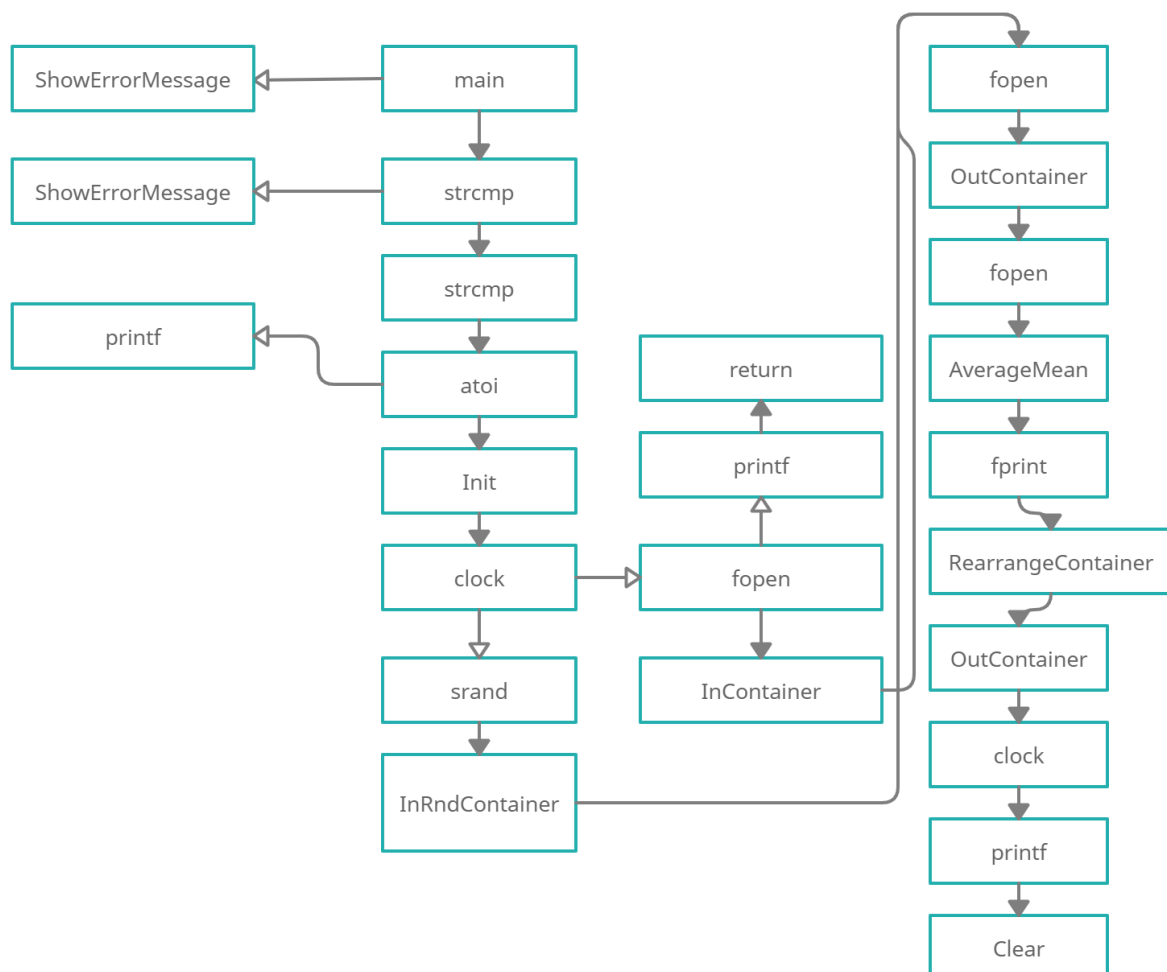
Stack	Память программы	
AverageMean	double AverageMean(...)	
Square		Память, Байт
	c: container	320004
	sum: double	8
Heap	i: int	4
c: container		
Глобальная память		
-		

Описание работы функции OutBall в рамках архитектуры:

Stack		Память программы
OutBall		void OutBall(...)
Square		Память, Байт
		ball: ball 16
		file: FILE 8
Heap		
-		
Глобальная память		
-		

Блок схема возможного стека, в результате работы функции main (с глубиной в 1 шаг):

Примечание: Пустая стрелочка обозначает то, что этот шаг не обязательно будет выполнен.



Основные характеристики программы:

Число header файлов: 6

Число source файлов: 6

Размер скомпилированное исполняемого файла: 52 Kb

Размер файлов с кодом:

main.cpp - 2,33 Kb

shape.cpp - 2,84 Kb

ball.cpp - 0,793 Kb

tetrahedron.cpp - 0,917 Kb

parallelepiped.cpp - 1,46 Kb

container.cpp - 3,06 Kb

tetrahedron.h - 0,904 Kb

ball.h - 0,777 Kb

parallelepiped.h - 0,976 Kb

container.h - 1,54 Kb

shape.h - 1,44 Kb

rnd.h - 0,567 Kb

Итого общий размер: 17,6 Kb

Суммарное количество строк кода: 560

Время выполнения для включенных в состав программы тестов:

тест1 - 7,11 ms (10 фигур на вход)

тест2 - 13,54 ms (59 фигур на вход)

тест3 - 33,42 ms (517 фигур на вход)

тест4 - 13,36 ms (24 фигуры на вход)

тест5 - 302,61 ms (9867 фигур на вход)

тест 6 - 124,71 ms (2671 фигур на вход)

рандомная генерация 5000 фигур - 139,79ms