Задание 1

Я скомпилировал программу example.c с отладочной информацией и без нее. Вот вывод утилиты valgrind:

Выдача с отладочной информацией

```
Invalid write of size 4
    at 0x10916B: f (example.c:6)
    by 0x109180: main (example.c:11)
    Address 0x4a9f068 is 0 bytes after a block of size 40 alloc'd
    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-
amd64-linux.so)
    by 0x10915E: f (example.c:5)
    by 0x109180: main (example.c:11)
...
40 bytes in 1 blocks are definitely lost in loss record 1 of 1
    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-
amd64-linux.so)
    by 0x10915E: f (example.c:5)
    by 0x109180: main (example.c:11)
```

Выдача без отладочной информации

```
Invalid write of size 4
    at 0x10916B: f (in /home/andrey05/Downloads/src/a.out)
    by 0x109180: main (in /home/andrey05/Downloads/src/a.out)
    Address 0x4a9f068 is 0 bytes after a block of size 40 alloc'd
    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-
linux.so)
    by 0x10915E: f (in /home/andrey05/Downloads/src/a.out)
    by 0x109180: main (in /home/andrey05/Downloads/src/a.out)
...
40 bytes in 1 blocks are definitely lost in loss record 1 of 1
    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-
linux.so)
    by 0x10915E: f (in /home/andrey05/Downloads/src/a.out)
    by 0x109180: main (in /home/andrey05/Downloads/src/a.out)
```

Разница в выводе заключается в том, что во втором случае valgrind дает более подробную информацию, включая номера строк и имена файлов, где произошли ошибки. Это может быть очень полезно для быстрого обнаружения и исправления проблем в коде.

Задание 2

В программе task_02.c ошибка связана с тем, что переменная y может быть использованна неинициализированной при вызове функции foo(y). Если условие

if (now == 0) не выполняется (т.е., now не равно 0), то переменной y не присваивается никакого значения, и она содержит неопределенное значение. Попытка передать это значение функции foo может привести к неопределенному поведению программы.

Я скомпилировал программу task_02.c и запустил программу под управлением утилиты valgrind. Выдача утилиты valgrind:

```
==83534== Conditional jump or move depends on uninitialised value(s)
==83534==    at 0x10917C: foo (task_02.c:6)
==83534==    by 0x1091C1: main (task_02.c:17)
```

Задание 3

В программе task_03_1.c в массиве x только 5 элементов, с индексами от 0 до 4. Попытка обратиться к элементу с индексом 5 приведет к выходу за пределы массива, что может привести к чтению данных из непредсказуемой области памяти, что может вызвать программный сбой.

В программе task_03_2.c программа пытается обратиться к элементу с индексом 5 в массиве x. Однако, массив x был выделен с помощью malloc на 5 элементов (с индексами от 0 до 4). Попытка доступа к элементу x[5] может привести к неопределенному поведению программы.

Без утилиты valgrind при запуске исполняемого файла первой программы выводится следующее сообщение:

```
*** stack smashing detected ***:
terminated
Aborted (core dumped)
```

При запуске второй не выводится ничего.

Выдача valgrind для первой программы:

```
ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Ошибка не была найдена

Утилита valgrind не помогает в поиске ошибок в программе task_03_1.c, потому что она не может обнаружить операции чтения или записи, выходящие за пределы диапазона, в массивы, выделенные статически или в стеке.

При запуске второй программы:

```
Invalid read of size 1
   at 0x1091EC: main (task_03_2.c:17)
   Address 0x4a9f045 is 0 bytes after a block of size 5 alloc'd
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x10919E: main (task_03_2.c:6)

Invalid write of size 1
   at 0x109210: main (task_03_2.c:19)
   Address 0x4a9f045 is 0 bytes after a block of size 5 alloc'd
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x10919E: main (task_03_2.c:6)
...
ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

Утилита успешно нашла две ошибки.

Задание 4

В программе task_04.c ошибка заключается в неверном выделении памяти для структуры. Вместо выделения памяти под структуру date, код выделяет память под указатель p, что неправильно.

Данная ошибка относится к логическим ошибкам.

Выдача утилиты valgrind:

```
Invalid write of size 4
   at 0x1091D5: create_date (task_04.c:23)
   by 0x109205: main (task_04.c:33)
 Address 0x4a9f048 is 0 bytes after a block of size 8 alloc'd
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091AF: create_date (task_04.c:18)
   by 0x109205: main (task_04.c:33)

Invalid read of size 4
   at 0x109215: main (task_04.c:36)
 Address 0x4a9f048 is 0 bytes after a block of size 8 alloc'd
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091AF: create_date (task_04.c:18)
```

```
   by 0x109205: main (task_04.c:33)
...
ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

Утилита подтверждает мой ответ. Она указывает на строки, в которых программа пытается заполнять указатель как структуру:

```
p->day = day;
p->month = month;
p->year = year;
```

Valgrind относит эту ошибку к категории Illegal read / Illegal write errors.

Задание 5

В программе task_05.c когда malloc вызывается во второй раз, указатель p перезаписывается на вновь выделенную память, не освободив предыдущую.

Данная ошибка относится к ошибкам утечки памяти.

Выдача утилиты valgrind:

```
4 bytes in 1 blocks are definitely lost in loss record 1 of 1
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091A6: main (task_05.c:11)
LEAK SUMMARY:
   definitely lost: 4 bytes in 1 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 0 bytes in 0 blocks
        suppressed: 0 bytes in 0 blocks
...
ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Утилита подтверждает мой ответ. Она указывает на строчку, в которй malloc переназначает указатель.

Valgrind относит эту ошибку к категории Memory leak detection.

Задание 6

В программе task_06.c ошибка заключается в том, что функция process выделяет память под int и присваивает ей значение n, но забывает освободить эту память перед возвращением значения.

Данная ошибка относится к ошибкам утечки памяти.

Выдача утилиты valgrind:

```
4 bytes in 1 blocks are definitely lost in loss record 1 of 1
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x109189: process (task_06.c:14)
   by 0x1091DD: main (task_06.c:31)

LEAK SUMMARY:
   definitely lost: 4 bytes in 1 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 0 bytes in 0 blocks
        suppressed: 0 bytes in 0 blocks

For lists of detected and suppressed errors, rerun with: -s
ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Утилита подтверждает мой ответ. Она указывает на строки, в которых программа выделяет память с помощью malloc, которую потом не освобождает.

Valgrind относит эту ошибку к категории Memory leak detection.

Задание 7

В программе task_07.c ошибка заключается в том, что указатель p не инициализирован, нужно выделить память для него с помощью malloc перед тем, как прочитать значение в него с помощью scanf.

Данная ошибка относится к логическим ошибкам(Wild pointer: использование непроинициализированного указателя).

Выдача утилиты valgrind:

```
Use of uninitialised value of size 8
   at 0x48DB1C9: __vfscanf_internal (vfscanf-internal.c:1896)
   by 0x48D61C1: __isoc99_scanf (isoc99_scanf.c:30)
   by 0x1091D7: main (task_07.c:14)

Invalid write of size 4
   at 0x48DB1C9: __vfscanf_internal (vfscanf-internal.c:1896)
   by 0x48D61C1: __isoc99_scanf (isoc99_scanf.c:30)
   by 0x1091D7: main (task_07.c:14)
 Address 0x0 is not stack'd, malloc'd or (recently) free'd
```

Утилита подтверждает мой ответ. Она указывает на строки, где используется непроинициализированная переменная.

Valgrind относит эту ошибку к категории Use of uninitialised or unaddressable values in system calls.

Задание 8

В программе task_08.c ошибка заключается в том, что указатель p используется повторно после освобождения памяти. Нужно заново выделить память для него с помощью malloc или других функций выделения памяти, прежде чем использовать его.

Данная ошибка относится к логическим ошибкам(Dangling pointer: использование указателя сразу после освобождения памяти).

Выдача утилиты valgrind:

```
Invalid write of size 4
   at 0x1091E0: main (task_08.c:20)
 Address 0x4a9f040 is 0 bytes inside a block of size 4 free'd
   at 0x484B27F: free (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091DB: main (task_08.c:18)
 Block was alloc'd at
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x10919E: main (task_08.c:11)

Invalid read of size 4
   at 0x1091EA: main (task_08.c:22)
```

```
 Address 0x4a9f040 is 0 bytes inside a block of size 4 free'd
   at 0x484B27F: free (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091DB: main (task_08.c:18)
 Block was alloc'd at
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x10919E: main (task_08.c:11)
```

Утилита подтверждает мой ответ. Она указывает на строки, где указатель используется повторно после освобождения памяти.

Valgrind относит эту ошибку к категории Illegal read / Illegal write errors.

Задание 9

В программе task_09.c ошибка заключается в том, что указатель pbeg, который вернула функция выделения памяти, изменяется.

Данная ошибка относится к логическим ошибкам (Изменение указателя, который вернула функция выделения памяти).

Выдача утилиты valgrind:

```
Invalid free() / delete / delete[] / realloc()
   at 0x484B27F: free (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x1091D5: main (task_09.c:23)
 Address 0x4a9f054 is 0 bytes after a block of size 20 alloc'd
   at 0x4848899: malloc (in
/usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x10918C: main (task_09.c:12)
...
LEAK SUMMARY:
   definitely lost: 20 bytes in 1 blocks
...

For lists of detected and suppressed errors, rerun with: -s
ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

Утилита подтверждает мой ответ. Она указывает, что область памяти была некорректно освобождена.

Valgrind относит эту ошибку к категории Illegal frees.