**[NOS(Nexon Open Studio) 과제]**

**주의사항 일반**

사용 언어는 C/C++로 한정.

결과물은 소스 코드 및 실행 파일(exe)과 함께 제출함.

소스 코드는 윈도우 환경에서 컴파일 가능해야 함.

실행 파일은 윈도우 32비트 환경에서 실행 가능해야 함.

STL(Standard Template Library)외 알고리즘/자료구조 해결을 위한 외부 라이브러리는 사용불가함.

제공된 소스코드(SimpleUtil.h/cpp)는 사용가능 함

**기둥을 피해 문서 수발실에 택배를 채우자! – 배경**

넥슨 문서수발실에는 매일같이 회사로 배달되는 택배로 정신이 없습니다. 한정된 공간에 다양한 규격의 택배를 채워야하는 까닭에, 택배를 차곡차곡 채우지 않으면 더 이상 택배를 받지 못할 상황이 벌어질 것만 같습니다. 게다가 문서 수발실 한가운데는 떡하니 둥근 기둥이 자리를 차지하고 있네요. 여러분은 문서수발실 직원을 도와 택배를 차곡차곡 채울 수 있는 방법을 강구하고, 택배를 효율적으로 배치한 결과를 출력해주도록 합니다. 물론, 더 많은 택배를 더 높은 효율로 받을 수록 좋은 평가를 받게 됩니다.

**문제**

input.txt 의 첫번째 줄에는 기둥의 중심 위치와 반지름, 그리고 색깔(red, green, blue)이 들어 있고, 다음 줄부터는 택배의 너비(width)와 높이(height), 그리고 색깔(red, green, blue)이 순서대로 한줄에 하나씩 들어 있습니다. 기둥은 평면도상에서 원(Circle)으로 표시되고 첫줄에 단 하나만 존재하며 고정되어 있지만, 택배들은 직사각형(Rectangle)로 표시되어 임의배치가 가능합니다.

input.txt (**아래 초록색 주석은 실제 포함되어 있지 않습니다.**)

|  |
| --- |
| 356 256 130 255 0 0 **// 기둥의 X 위치, Y 위치, 반지름, red, green, blue**  27 9 135 233 217 **// 택배의 width, height, red, green, blue**  16 45 238 214 190 **// …**  33 31 242 100 114 **// …**  24 14 228 175 131  ........ |

이 파일을 한줄씩 읽어가면서 기둥 및 택배 정보를 입력받아 512 x 512 크기(너비 x 높이)의 공간에 채워넣습니다. 물론, 먼저 둥근 기둥을 배치하고, 이어서 택배를 놓아야겠죠. 채워나가는 순서를 뒤바꾸는 것은 허용되지 않지만 (즉, 입력 순서의 변경/정렬(Sorting) 등은 허용되지 않음. 재배치 허용하지 않음) 입력 받은 택배의 너비와 높이를 뒤바꾸는 것은 허용됩니다(회전 허용)

채워넣어야하는 택배의 부피는 항상 문서수발실의 공간보다 많기에(즉, input.txt 에서 택배 부피합 > 512 x 512) 받을 수 있는 만큼 최대한 택배를 받아, 효율적으로 512 x 512 공간내에서 기둥을 피해 순서대로 채워나가는 것이 중요합니다. 더 많은 입력을 받을 수록, 더 높은 점수를 받게 됩니다.

기둥과 택배를 512x512 공간에 모두 채운 후에는 해당 정보를 (1) 512x512크기의 비트맵이미지(output.bmp)로 출력하고, 다음과 같이 (2) 배치로그(log)를 output.txt 에 기록해서 출력하도록 합니다. (배치로그에는 기둥정보는 기록하지 말것) 배치로그의 맨 마지막 줄에는 공간효율((사용된 택배의 총 넓이)/(수발실 넓이==512\*512))을 표시합니다.

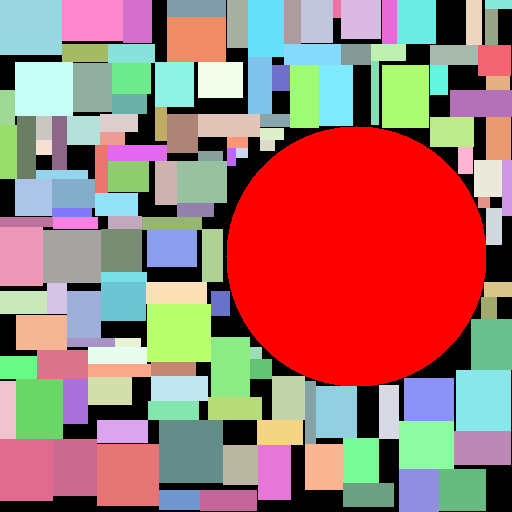
|  |
| --- |
| 1:485,0,27,9,243 **// 택배순서:X위치,Y위치,너비,높이,넓이**  2:466,0,16,45,963  ……  135:63,379,25,45,166551  136:343,483,51,24,167775  공간 효율 : 0.640011 **//공간효율 == (167775)/(512\*512)** |

채점은 제공된 input.txt 과 함께 별개의 채점용 input.txt 에 의해서도 평가되며, 실행 파일의 실행시간은 최대 10초를 넘어서는 안됩니다. 실행 도중 오류가 발생하거나, 시간제한을 어기면 감점 혹은 오답처리될 수 있습니다. 따라서 input.txt 를 읽어서, output.txt 를 만들어 내는 exe 파일을 반드시 첨부해야한 합니다.

**제출결과 예**

\* 아래결과는 참고결과/input.txt 를 읽어들였을 때, 출력될 수 있는 결과예 중 하나입니다.

(1) output.bmp



(2) output.txt

|  |
| --- |
| 1:485,0,27,9,243  2:466,0,16,45,963  3:478,45,33,31,1986  4:486,76,24,14,2322  ……  131:159,420,64,63,161242  132:223,445,35,40,162642  133:148,401,51,19,163611  134:258,445,33,55,165426  135:63,379,25,45,166551  136:343,483,51,24,167775  공간 효율 : 0.640011 |

정리해서, (1) 소스코드(예, cpp,h) 및 빌드파일(예, vcxproj, sln), (2) 실행가능한 exe 파일 (3) 결과물(output.txt, output.bmp)을 제출해야합니다. 구체적으로 결과물의 경우, [02. 채점입력] 의 [입력1] 과 [입력2]에 들어있는 각각의 input.txt 에 대한 output.txt, output.bmp 를 제출부탁드립니다.

**제공된 소스코드**

SimpleUtil.h/cpp 에는 Win32/C++ 환경에서 사용가능한 세개의 API가 제공됩니다. SimpleSaveBMP() 의 경우, 버퍼(pData) 에 기록된 raw 이미지 정보를 단순한 형태의 BMP 파일로 출력해주는 기능을 하고, FillRect()는 BMP로 출력될 버퍼의 특정 위치에 원하는 색깔의 사각형(Rectangle)을 기록해주는 역할을 하며, FillCircle()는 비슷하게 원(Circle)을 주어진 색깔로 채우는 역할을 합니다. 자세한 용법 및 간단한 사용 예제는 SimpleUtil.h/cpp 를 참고합니다.