

Python	<p>Tytuł: Keyboard</p> <p>Treść:</p> <p>Zrób program, który na naciśnięcie konkretnego klawiszu na klawiaturze będzie odgrywał konkretny dźwięk</p> <p>Wymagania:</p> <ul style="list-style-type: none">• Użyty język: Python• Użycie bibliotek dozwolone• Wynikiem ma być program, który będzie reagował na wciśnięcie klawiszu odgrywaniem dźwięku. Każdy klawisz powinien wydawać inny dźwięk (nie musi być tego słychać, ale program ma mieć dla każdego klawiszu inną wartość). Klawisze funkcyjne (Alt, Ctrl, Shift, Esc...) nie muszą być obsługiwane.																																
Pseudo-assembler	<p>Tytuł: Obliczanie współrzędnych środka linii</p> <p>Treść: Używając algorytmu Bresenhama (zakładamy, że poruszamy się tylko w pierwszej ósemce układu współrzędnych – kąty: 0 – 45) oblicz współrzędne środka linii.</p> <p>Specyfikacja</p> <p>Wejście: załóż, że współrzędne linii są podane w rejestrach odpowiednio: $(R_0, R_1) - p_1 = (x_1, x_2)$; $(R_2, R_3) - p_2 = (x_2, y_2)$</p> <p>Wyjście: wynik umieść w rejestrach odpowiednio: (R_{14}, R_{15})</p> <p>Do oddania jest skrypt w języku Pseudoasemblera Politechniki Warszawskiej, który jest poprawnie interpretowany przez internetową wersję Interpretera PPW KNI.</p>																																
Algorytmiczne	<p>Tytuł: Oglądanie klocków</p> <p>Treść:</p> <p>Dana jest plansza $n \times n$. Na każdym polu planszy może stanąć klocek o wysokości h (całkowite, $1 \leq h \leq n$). Klocki stawiamy dokładnie na środku pola i nie rozpatrujemy kąta widzenia, czyli zakładamy, że zza klocka o wysokości większej klocek o wysokości mniejszej nie będzie widoczny. Niektóre kolumny i wiersze planszy zawierają wskazówkę, ile klocków widać z danego miejsca, w przypadku jej braku liczba ta jest dowolna. Dla przykładu:</p> <div><div><div>3</div><div>1</div><table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table><div>3</div><div>4</div></div><div><div>3</div><div>1</div><table><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr><tr><td>3</td><td>4</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>4</td><td>3</td><td>1</td><td>2</td></tr></table><div>3</div><div>4</div></div></div> <p>Wyjaśnienie:</p>																	2	1	4	3	3	4	2	1	1	2	3	4	4	3	1	2
2	1	4	3																														
3	4	2	1																														
1	2	3	4																														
4	3	1	2																														

	<ul style="list-style-type: none"> • W pierwszej kolumnie od góry widoczne są trzy klocki. Pierwszy ma wysokość 2 i jest widoczny, drugi ma wysokość 3 i jest widoczny, trzeci ma wysokość 1 i jest niewidoczny (zakryty przez poprzednie), czwarty ma wysokość 4 i jest widoczny. • W drugiej kolumnie brakuje wskazówki, klocki zostały ułożone w dowolnej kolejności • W drugim wierszu od prawej są widoczne trzy klocki. Pierwszy ma wysokość 1 i jest widoczny, drugi ma wysokość 2 i jest widoczny, trzeci ma wysokość 4 i jest widoczny, czwarty ma wysokość 3 i jest niewidoczny (zakryty przez poprzedni) <p>Dane wejściowe</p> <p>Każde uruchomienie programu obsługuje jeden przypadek testowy. Wprowadzane są 4 linie, zawierające kolejno wskazówki zgodnie z ruchem wskazówek zegara, czyli kolumny od góry, wiersze od prawej, kolumny od dołu, wiersze od lewej, ale wartości podajemy od lewej do prawej, z góry na dół. Kolejne cyfry oddzielone są spacjami, brak wskazówki symbolizowany jest znakiem 0. Np.</p> <pre>3 0 1 0 0 3 0 0 0 0 0 0 0 0 4 0</pre> <p>Dane wyjściowe</p> <p>Na wyjściu aplikacji powinno pojawić się rozwiązanie danego problemu, gdzie kolejne linie wypisane są w kolejnych wierszach, a wartości kolumn oddzielone spacjami lub komunikat „BRAK ROZWIĄZAN”. Np.</p> <pre>2 1 4 3 3 4 2 1 1 2 3 4 4 3 1 2</pre> <p>Wymagania:</p> <ul style="list-style-type: none"> • Algorytm powinien liczyć jeden przypadek testowy krócej niż minutę • Jeśli dla danych wejściowych jest możliwe kilka rozwiązań wystarczy wypisanie jednego • Jeśli dla danych wejściowych nie ma rozwiązań, należy to wskazać • Dowolny język programowania • Możliwość wykorzystania zewnętrznych bibliotek, o ile nie rozwiązują one same wspomnianego problemu • Załóż, że $n < 10$ • Wszystkie wartości są naturalne <p>Źródło: http://home.elka.pw.edu.pl/~ewarchul/wae/#zadanie-26</p>
Nieszablonowe	Temat: Open source

	<p>Treść: Dołączcie do dowolnego projektu open source robiąc co najmniej jednego commita. Swoje dokonania udokumentujcie opisując co zrobiliście oraz dodając linka do wybranego przez was projektu. Przykłady projektów przyjmujących początkujących możecie znaleźć tutaj: https://github.com/MunGell/awesome-for-beginners</p>
Otwarte	<p>Temat: Floating objects</p> <p>Treść: Napisz PRZEPIĘKNE, dynamiczne tło do stron internetowych - swobodnie poruszające się kulki (można przyjąć również inne kształty).</p> <p>Wymagania:</p> <ul style="list-style-type: none"> • zaczynamy z planszą losowo rozłożonych kształtów poruszających się w losowym kierunku z losową prędkością; • przy naciśnięciu kształtu myszką, powinno się coś zmienić: kierunek / prędkość (można zaproponować swój pomysł); • mile widziane efekty hoverowe; • Kształty poruszają się płynnie <p>* ad. pkt. 1.: Można zaproponować inne rozmieszczenia kul niż losowe, mogą być to jakieś regularne wzory itp. (każdy uzasadniony i estetyczny pomysł zostanie pozytywnie oceniony). Ruch kształtów, natomiast, musi zaczynać się z losowego miejsca.</p> <p>Oceniane:</p> <ul style="list-style-type: none"> • spełnienie powyższych wymagań; • zastosowanie odpowiednich algorytmów / wzorów / technologii w celu uzyskania płynnych animacji - brak schodkowania, zacinania się i wolnego ładowania grafik! • ze względu na to, że jest to "algorytmiczna" ścieżka rozwoju, pod uwagę brany jest bardziej kod pod spodem niż ostateczny efekt wizualny. Można np. skorzystać z gotowych palet kolorystycznych z material.io. Nie zapominajmy jednak, że jest to zadanie frontendowe i całość musi jakoś wyglądać! • staranność i przejrzystość w zapisie kodu, komentarze w miejscach, gdzie są one potrzebne