

**EGZAMIN MATURALNY  
OD ROKU SZKOLNEGO 2014/2015**

**INFORMATYKA  
CZEŚĆ I  
POZIOM ROZSZERZONY**

**ROZWIĄZANIA ZADAŃ I SCHEMATY PUNKTOWANIA  
(A1,A2, A3, A4, A7)**

**GRUDZIEŃ 2014**

Numer zadania	Numer podpunktu	Oczekiwana odpowiedź	Maksymalna punktacja za część zadania	Maksymalna punktacja za zadanie
1	1.1.	Za podanie prawidłowej odpowiedzi PPF – <b>1 punkt</b> .	1	5
	1.2.	Za prawidłowo napisany algorytm – <b>2 punkty</b> . Przykładowe rozwiązanie w języku C++:  <pre>int n=0; do{     cyfry[n]=k%10;     k=k/10;     n++; } while (k!=0); cout &lt;&lt; n;</pre>	2	
	1.3.	Za prawidłowo napisany algorytm – <b>2 punkty</b> . Przykładowe rozwiązanie w postaci listy kroków: (1) $j \leftarrow 0$ ; (2) powtarzaj (3) $k \leftarrow k\text{-potega}(\text{cyfry}[j],n)$ ; (4) $j \leftarrow j+1$ ; (5) dopóki ( $j < n$ ); (6) jeżeli ( $k=0$ ) pisz PRAWDA przeciwnym wypadku pisz FAŁSZ	2	
2	2.1.	Za podanie prawidłowej odpowiedzi PFP – <b>1 punkt</b> .	1	4
	2.2.	Za podanie prawidłowej odpowiedzi FFP – <b>1 punkt</b> .	1	
	2.3.	Za podanie prawidłowej odpowiedzi PFPF – <b>1 punkt</b> .	1	
	2.4.	Za podanie prawidłowej odpowiedzi PFFF – <b>1 punkt</b> .	1	

	3.1.	<p>Za poprawne uzupełnienie obu wierszy – <b>1 punkt</b>:</p> <ul style="list-style-type: none"> <li>• pierwszy wiersz (01101110 01000000),</li> <li>• drugi wiersz (KAJAK)</li> </ul>	1	
3	3.2.	<p>Za poprawny algorytm zliczania różnych znaków – <b>2 punkty</b>, w tym za:</p> <ul style="list-style-type: none"> <li>• poprawne testowanie czy znak jeszcze nie wystąpił, lub zapis nowego znaku do zbioru – <b>1 punkt</b>,</li> <li>• poprawne określenie liczebności zbioru różnych znaków – <b>1 punkt</b>.</li> </ul> <p><b>Przykładowe rozwiązanie 1:</b></p> <pre>string s = "AAAABBBCCCBBAACC"; int n = s.length(); char R[100]; int r=0; for (int i=0; i&lt;n; i++) {     char c = s[i];     int j=0;     while (j&lt;r &amp;&amp; R[j]!=c)         j++;     if (j==r)     {         R[j] = c;         r++;     } } cout&lt;&lt;r&lt;&lt;endl;</pre> <p><b>Przykładowe rozwiązanie 2:</b></p> <pre>wymaga: #include &lt;set&gt; string s = "AAAABBBCCCBBAACC"; set &lt;char&gt; R; set &lt;char&gt; :: iterator it; for (int i=0; i&lt;s.length(); i++)     R.insert(s[i]); cout&lt;&lt;R.size()&lt;&lt;endl;</pre>	2	6

	3.3.	<p>Za poprawny algorytm dekodowania znaków – <b>3 punkty</b>, w tym za:</p> <ul style="list-style-type: none"> <li>• poprawne powiązanie testowania bitu w bajcie skompresowanym z ustawieniem bitu w kodzie znaku – <b>1 punkt</b>,</li> <li>• poprawną obsługę indeksowania bitów w bajcie skompresowanym (od 0 do 8) – <b>1 punkt</b>,</li> <li>• poprawną detekcję znaku przypisanego do kodu – <b>1 punkt</b>,</li> </ul> <p><b>Przykładowe rozwiązanie:</b></p> <pre> cout&lt;&lt;"\nodszyfrowanie: \n"; int i=0; // numer kolejnego bajtu skompresowanego int bajt; // bajt skompresowany int ib=0; // indeks bitu w bajcie skompresowanym int kod; // kod znaku int ik=0; // indeks bitu w kodzie znaku while (i&lt;n) {     bajt=V[i]; // pobierz bajt tekstu skompresowanego     ib=0; // indeks bitu w powyższym bajcie     kod=0;     while (ik&lt;bity)     {         // jeżeli bajt ma ustawiony bit <i>ib</i>, to ustaw bit <i>ik</i>         w kodzie         if ((bajt &amp; (1&lt;&lt;ib))) kod = kod   (1&lt;&lt;ik);         ik++;         if (ik==bity)         { // obsłużyłeś wszystkie bity kodu, więc         odzyskaj w tablicy V znak przypisany do aktualnego         kodu, jeżeli znajdziesz, to wypisz znak na         standardowe wyjście             int j=0;             while ( kod!=R[j].kod &amp;&amp; j&lt;r) j++;             // R – dana tablica par: {znak, kod}             if (j&lt;r) cout&lt;&lt; R[j].znak;             // r – dana liczba różnych znaków             ik=0; // wyzeruj kod oraz indeks bitu w kodzie             kod=0;         }         ib++;         if (ib==8) break;     }     i++; } </pre>	3	
--	------	--	---	--