

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

**INFORMATYKA
CZĘŚĆ 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 – 1 punkt.	1	5
	1.2.	Za prawidłowo napisany algorytm – 2 punkty. Przykładowe rozwiązanie w języku C++: <pre>int n=0; do{ cyfry[n]=k%10; k=k/10; n++; } while (k!=0); cout << n;</pre>	2	
	1.3.	Za prawidłowo napisany algorytm – 2 punkty. 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 – 1 punkt.	1	4
	2.2.	Za podanie prawidłowej odpowiedzi FFP – 1 punkt.	1	
	2.3.	Za podanie prawidłowej odpowiedzi PFPP – 1 punkt.	1	
	2.4.	Za podanie prawidłowej odpowiedzi PFFF – 1 punkt.	1	

3	3.1.	Za poprawne uzupełnienie obu wierszy – 1 punkt: • pierwszy wiersz (01101110 01000000), • drugi wiersz (KAJAK)	1	6
	3.2.	Za poprawny algorytm zliczania różnych znaków – 2 punkty, w tym za: • poprawne testowanie czy znak jeszcze nie wystąpił, lub zapis nowego znaku do zbioru – 1 punkt, • poprawne określenie liczebności zbioru różnych znaków – 1 punkt. Przykładowe rozwiązanie 1: <pre> string s = "AAAABBBCCCBBAACC"; int n = s.length(); char R[100]; int r=0; for (int i=0; i<n; i++) { char c = s[i]; int j=0; while (j<r && R[j]!=c) j++; if (j==r) { R[j] = c; r++; } } cout<<r<<endl; </pre> Przykładowe rozwiązanie 2: <pre> wymaga: #include <set> string s = "AAAABBBCCCBBAACC"; set <char> R; set <char> :: iterator it; for (int i=0; i<s.length(); i++) R.insert(s[i]); cout<<R.size()<<endl; </pre>	2	

	3.3.	Za poprawny algorytm dekodowania znaków – 3 punkty, w tym za: • poprawne powiązanie testowania bitu w bajcie skompresowanym z ustawieniem bitu w kodzie znaku – 1 punkt, • poprawną obsługę indeksowania bitów w bajcie skompresowanym (od 0 do 8) – 1 punkt, • poprawną detekcję znaku przypisanego do kodu – 1 punkt, Przykładowe rozwiązanie: <pre> cout<<"\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<n) { bajt=V[i]; // pobierz bajt tekstu skompresowanego ib=0; // indeks bitu w powyższym bajcie kod=0; while (ik<bity) { // jeżeli bajt ma ustawiony bit <i>ib</i>, to ustaw bit <i>ik</i> // w kodzie if ((bajt & (1<<ib))) kod = kod (1<<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 && j<r) j++; // R – dana tablica par: {znak, kod} if (j<r) cout<< 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	
--	------	--	---	--