CENTRALNA
KOMISJA
EGZAMINACYJNA

# 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**.<br>Przykładowe rozwiązanie w języku C++:<br>`int n=0;`<br>`do{`<br>`  cyfry[n]=k%10;`<br>`  k=k/10;`<br>`  n++;`<br>`} while (k!=0);`<br>`cout << n;` | 2 | |
| | 1.3. | Za prawidłowo napisany algorytm – **2 punkty**.<br>Przykładowe rozwiązanie w postaci listy kroków:<br>(1) j ← 0;<br>(2) powtarzaj<br>(3) k ← k-potega(cyfry[j],n);<br>(4) j ← j+1;<br>(5) dopóki (j<>n);<br>(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 PFPF – **1 punkt**. | 1 | |
| | 2.4. | Za podanie prawidłowej odpowiedzi PFFF – **1 punkt**. | 1 | |

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

| | | | | |
|---|---|---|---|---|
| | 3.3. | Za poprawny algorytm dekodowania znaków – **3 punkty**, w tym za:<br>• poprawne powiązanie testowania bitu w bajcie skompresowanym z ustawieniem bitu w kodzie znaku – **1 punkt**,<br>• poprawną obsługę indeksowania bitów w bajcie skompresowanym (od 0 do 8) – **1 punkt**,<br>• poprawną detekcję znaku przypisanego do kodu – **1 punkt**,<br><br>**Przykładowe rozwiązanie:**<br>cout<<"\nodszyfrowanie: \n";<br>int i=0; // numer kolejnego bajtu skompresowanego<br>int bajt; // bajt skompresowany<br>int ib=0; // indeks bitu w bajcie skompresowanym<br>int kod; // kod znaku<br>int ik=0; // indeks bitu w kodzie znaku<br>while (i<n)<br>{<br>bajt=V[i]; // pobierz bajt tekstu skompresowanego<br>ib=0;// indeks bitu w powyzszym bajcie<br>kod=0;<br>while (ik<bity)<br>{<br>// jezeli bajt ma ustawiony bit *ib*, to ustaw bit *ik* w kodzie<br>if ((bajt & (1<<ib))) kod = kod \| (1<<ik);<br>ik++;<br>if (ik==bity)<br>{ // obsłużyłeś wszystkie bity kodu, więc odszukaj w tablicy *V* znak przypisany do aktualnego kodu, jeżeli znajdziesz, to wypisz znak na standardowe wyjscie<br>int j=0;<br>while ( kod!=R[j].kod && j<r) j++;<br>// R – dana tablica par: {znak, kod}<br>if (j<r) cout<< R[j].znak;<br>// r – dana liczba różnych znaków<br>ik=0; // wyzeruj kod oraz indeks bitu w kodzie<br>kod=0;<br>}<br>ib++;<br>if (ib==8) break;<br>}<br>i++;<br>} | 3 | |