

1.1 FFPF

1.2 FFPF1.3 PFFF1.4 FFPF

2.1

84 | 5 | 50,20,10,2,2

533 | 7 | 200,200,100,20,10,2,1

2.2

```
int bank =0;
    for (int j = 0; j < Nomin.length; j++) {
        while(K>=Nomin[j]) {
            bank=bank+K/Nomin[j];
            K=K%Nomin[j];
        }
    }
    return bank;
```

2.3

indeksy P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
wartości P dla nominału 2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7

indeksy P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
wartości P dla nominału 7	0	1	1	2	2	3	3	1	2	2	3	3	4	4	2

indeksy P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
wartości P dla nominału 10	0	1	1	2	2	3	3	1	2	2	1	2	2	3	2

3.1

92 | - | 1,2,4,23,46,92 | Nie

114 | 2 · 3 · 19 | 1,2,3,6,19,38,57,114 | Tak

3.2

```
public static String pierwsza (int a) {
    int dzielniki=0;
    for (int i = 2; i < a; i++) {
        if(a%i==0) {
            return "nie";
        }
    }
    return "tak";
}
```

3.3

```
public static String sfeniczna (int a) {
    int liczba_pierwszych=0;
    int [] dzielniki=new int[a];
    for (int i = 2; i < a; i++) {
        if(pierwsza(i).equals("tak")) {
```

```

        dzielniki[liczba_pierwszych]=i;
        liczba_pierwszych++;
    }

}
int temp =0;
for (int i = 0; i < dzielniki.length; i++) {
    for (int j = 1; j < dzielniki.length; j++) {
        for (int j2 = 2; j2 < dzielniki.length; j2++) {
            temp = dzielniki[i]*dzielniki[j]*dzielniki[j2];
            if(temp == a) {
                return dzielniki[i]+
""+dzielniki[j]+"*"+dzielniki[j2];
            }
        }
    }
}
return "nie";
}

```