

## Kolokwia 2017 - relacja

Czwartek 11.00:

`unsigned long check_div(long a, long b, long c);`

Jeśli c równe zero to

a) zwraca 1 jeśli a podzielne przez b

b) zwraca 0 jeśli a niepodzielne przez b.

Jeśli c różne od 0 to dzieli a przez liczby z zakresu 1..64 i ustawia odpowiednio bity np. jeśli a podzielne przez 1 to najmłodszy bit równy 1, jeśli a niepodzielne przez 2 to kolejny bit równy 0 itd.

Czwartek 12.30:

`unsigned int check_tab(int* tab, int n, int* max)`

Na wejściu tablica n elementów z zewnątrz; sprawdzić elementy tablicy pod kątem występowania wartości od 0 do 63, następnie policzyć ile razy któryś tam element się pojawił, np 0 pojawiło się ile razy, dwójka ile razy (zliczamy wystąpienia), poprzez wskaźnik przekazany w trzecim argumencie zwrócić maksimum wystąpień; Funkcja ma zwrócić wartość w postaci 64 bitów - każdy bit określa wystąpienie liczby odpowiadającej temu bitowi, np jeśli była dwójka w tablicy to ma się pojawić jedynka na pozycji 3, ostatni bit odpowiada za liczbę 63 (wynik funkcji - które liczby się pojawiły, maks mówi o tym ile razy pojawiła się najczęściej występująca liczba).

Piątek 9:30

`char* encode(char* buf, unsigned int mask, int operation, int character);`

Ogólnie trzeba było zrobić funkcję która zmieni cyfry w stringu (tylko te gdzie maska ma bit ustawiony na 1) w sposób zgodny z podaną operacją.

Buf to był wskaźnik na stringa, w masce liczyło się tylko 10 najmłodszych bitów, każdy bit oznaczał czy wykonać operację na danej cyfrze, np. jak tylko najmłodszy bit był 1 to zmieniamy tylko cyfrę 0. Operacja mogła wynosić od 0 do 3 (jak się potem w testach okazało może być jakakolwiek i wtedy powinna zostać obsłużona jako 0, nie pamiętam żeby na kolosie to mówił).

Dla 0 nie robimy nic.

Dla 1 usuwamy cyfrę

Dla 2 robimy jak on to nazwał "transpozycję" - zamieniamy 0 na 9, 1 na 8, 2 na 7 itp

Dla 3 podmieniamy cyfrę znakiem przekazanym jako 4 argument