

УТВОРЕННЯ І РОЛЬ ЯДЕР ${}^8\text{Be}$ ПРИ ФРАГМЕНТАЦІЇ ЛЕГКИХ ЯДЕР

Лясковець Сергій

Київський національний університет
імені Тараса Шевченка
фізичний факультет

19 квітня 2018 р.

Огляд тематики

У процесах кластеризації ядерного речовини в легких ядрах особлива роль належить ядру ^8Be . Через цей стан утворюється помітна частка α частинок - фрагментів всіх релятивістських ядер. В даній статті приділено особливу увагу експериментальним особливостям виділення ядер ^8Be при фрагментації релятивістського ядра ^{10}B як найлегшого ядра, в якому цей кластер може утворитися. Розраховані ймовірності утворення ядер ^8Be при фрагментації релятивістських ядер ^{10}B , ^{12}C , ^{16}O і проведено порівняння цих розрахунків з експериментом. Показано, що автори статті, в основному, правильно розуміють механізм утворення цього кластера при фрагментації релятивістських ядер, так як розрахунки, проведені на основі цих уявлень, підтверджуються експериментом.

-
-
-

Реалізація

```
int main()
{
    //unsigned int start_time = clock(); // начальное время
    // здесь должен быть фрагмент кода, время выполнения которого нужно измерить

    long int N, Q = 0 , p = 1;
    cin >> N; // вводим число
    if(N==0) // якщо воно 0, Q=10
    {
        cout<<"10"<<endl;
        return 0;
    }
    if(N<10) // якщо число з 1 цифри, виводимо саме це число
    {
        cout<<N<<endl;
        return 0;
    }
}
```

Рис. 1 : Використаний код

Реалізація

```
// для інших чисел алгоритм наступний: знаходимо на кожному кроці ітерації найбільший дільник числа, дільник складається з 1 цифри!  
// запонуємо число Q цими дільниками у зворотньому порядку, аби число Q було найменше  
for(int div = 9; div > 1; --div)  
{  
    while( (N%div) == 0)  
    {  
        Q += p * div;  
        p = p * 10;  
        N /= div;  
    }  
}  
// якщо після цих кроків, результат ділення на всі одноцифрові дільники більше 1, значить число розкладається на дільники більше 9,  
// наприклад, 44=4*11, в нас же вимагається щоб дільники складались з 1 цифри  
if(N>1)  
{  
    cout<<"-1"<<endl;  
    return 0;  
}  
// виводимо результат  
cout << Q << endl;  
// unsigned int end_time = clock(); // конечное время  
// unsigned int search_time = end_time - start_time; // искомое  
// cout<<search_time;  
return 0;  
}
```

Рис. 2 : Використаний код