اعضای گروه:

بهشاد باغبان زاده، شکیبا شهیدیانی

کد را به صورت مشترک از طریق نرم افزار anydesk پیش بردیم.

هیچ قسمتی از کد از اینترنت کپی نشده حتی شبه کد هم نداشتیم. تماما از صفر تا صد توسط خود دانشجو نوشته شده است. این کد الگوریتم LZ78 را پیاده سازی می کند. در فایل های فرستاده شده سه قسمت وجود دارد. Coding.m،main.m ،MakeDic.m

در قسمت coding.m ، Function اصلی وجود دارد.

InputString=’ AABABBBABAABABBBABBA’

dictv={'A','B'};

ورودی ها هستند(به طورمثال)که شامل یه دیکشنری از حروفی که در رشته وجود دارد و رشته ی مورد نظر است.

و خروجی این برنامه :

یک دیکشنری باینری از دیکشنری اولیه و یک آرایه به صورت باینری می باشد.

الگوریتم LZ78:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| ABBA | BB | ABAB | ABA | B | ABB | AB | A |
| 3A | 4B | 5B | 2A | 0B | 2B | 1B | A |

الگوریتم به این شکل است که اول روی رشته پیش میرویم و هر بخش را جدا میکنیم.  
بعد روی هر بخش پیش میرویم حرف اخر رو مینویسیم و به جای حروف سمت چپ شماره ایندکس بخش های جدا شده از جداسازی مرحله قبل را جایگزین میکنیم.  
حال به بخش کاراکتری یک عدد اختصاص میدهیم(MakeDic).  
و اعداد را هم به صورت باینری ذخیره میکنیم.

در فایل های ارسال شده بخش کامنت شده مشخص شده است که قسمت ابتدایی کد سطر دوم و سوم را میسازند و از آنجا به بعد برای باینری کردن سطر سوم است.تابع makeDic برای باینری کردن Dictv است که با استفاده از آن بتوان قسمت حروف سطر سوم را باینری کرد. در قسمت اخر حروف و اعداد جدا شده و با استفاده از dicB که خروجی MakeDic بود باینری میشوند. نکته اینجاست که اعداد موجود باید طبق فرمول 2^j-1 ادرس j بیتی تبدیل به باینری شوند. یعنی تعداد بیت هایشان تغییر میکند.سپس در هر مرحله بخش حروف و اعداد به یکدیگر کانکت می شوند.

خروجی :

0111010010100101110010110

dicB={0,1}