فصل سوم : spell correction

۱-الف) سه مورد از کاربردهای استفاده از wildcard query را به همراه مثال بنویسید.

پاسخ :

۱. کاربر در املای یک کلمه از کوئری شک داشته باشد. مثال :
Sydney vs. Sidney, which leads to the wildcard query S*dney
۲. کاربر بداند یک کلمه با چند املای مختلف نوشته می‌شود و بخواهد در جستجوی اسناد تمامی حالت‌های کلمه‌ی کوئری، برگردانده شود. مثال :
color vs. colour, which leads to the wildcard query col*r
۳. کاربر به دنبال اسنادی باشد که شکل های مختلف یک کلمه را در بر دارند و نمیداند که موتور جستجو ریشه‌یابی (stemming) را انجام میدهد یا خیر. مثال :
judicial vs. judiciary, leading to the wildcard query judicia*
۴. کاربر از بیان صحیح یک کلمه یا عبارت خارجی مطمئن نباشد. مثال :
the query Universit* Stuttgart
(universitat in German - university in English)

*به هرگونه مثال یا کاربرد صحیح به جز موارد بالا نیز نمره‌ی کامل تعلق می‌گیرد.

ب) سه روش زیر که در پردازش wildcard queris استفاده میشود را از نظر سرعت پردازش، قدرت پاسخگویی به کوئری‌های مختلف و حافظه مصرفی مقایسه کنید.

I. B-tree
II. permuterm index
III. bigram index

پاسخ :

سرعت پردازش : permuterm index > B-tree > bigram index

قدرت پاسخگویی : bigram index > permuterm index > B-tree

حافظه مصرفی : bigram index ~ permuterm index > B-tree

ج) فرض کنید یک لغت نامه شامل کلمات زیر است:

From,romantic,room,chrome, agronomy

نشان دهید این کلمات در دو حالت permuterm index , bigram index چگونه ذخیره می‌شوند. سپس کوئری *rom* و *rom را در هر دو حالت پردازش کنید. تمامی جزئیات را بنویسید.

پاسخ :

الف) روش bigram index :

$f => from

Fr => from

Ro => agronomy, chrome , from , romantic , room

Om => agronomy, chrome , from , romantic , room

$r => romantic , room

Ma =>romantic

An => romantic

nt =>romantic

ti => romantic

ic => romantic

oo =>room

$c => chrome

ch => chrome

hr => chrome

me => chrome

$a => agronomy

ag => agronomy

gr => agronomy

on => agronomy

no => agronomy

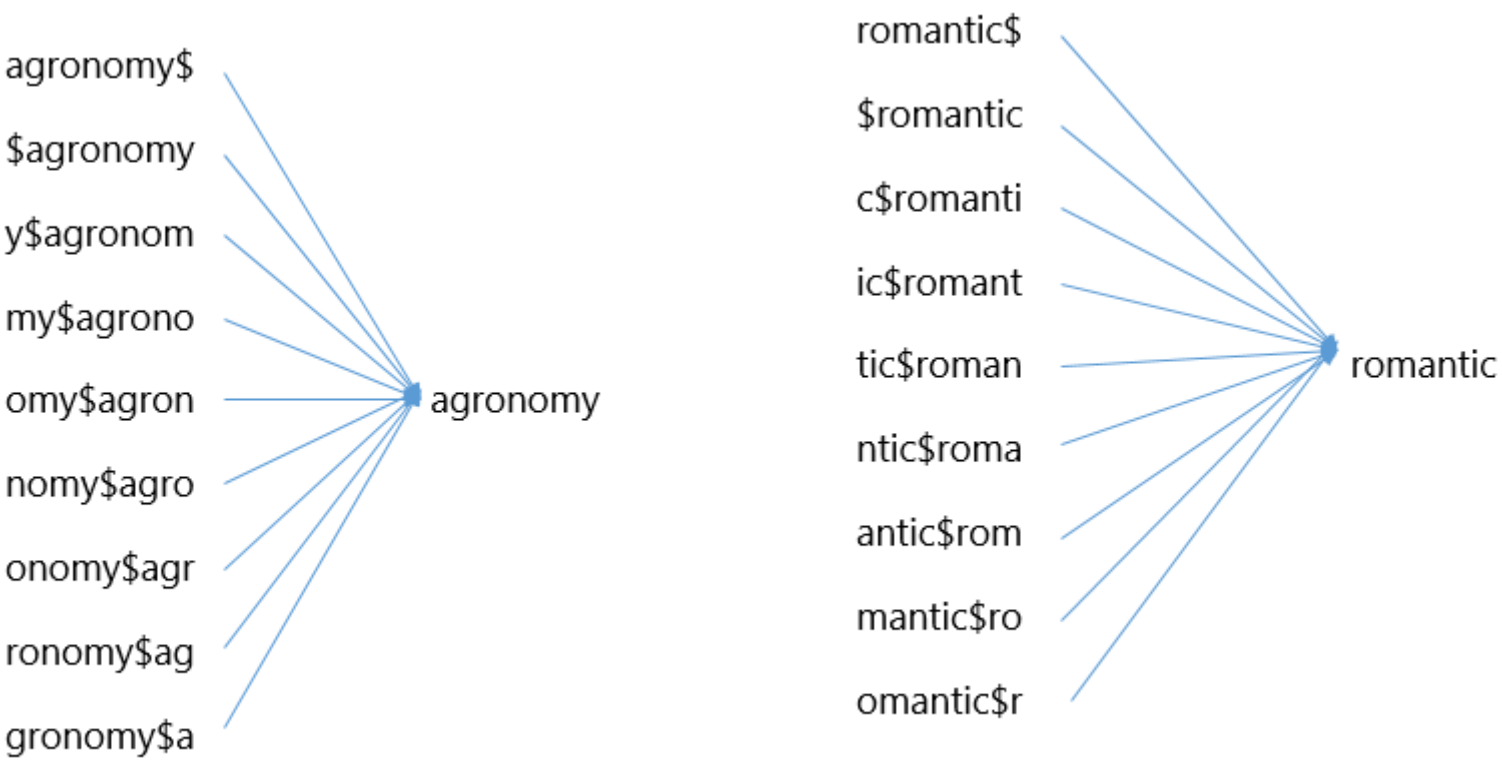
my => agronomy

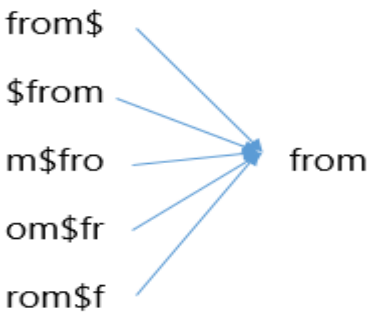
برای کوئری *rom* عبارت ro and om جستجو می‌شود و حاصل جستجو :

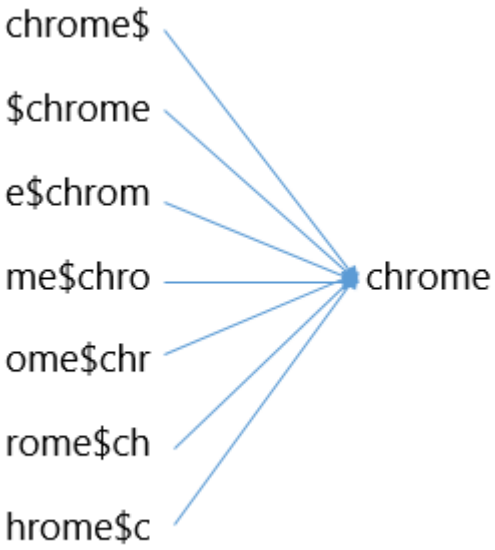
agronomy, chrome , from , romantic , room

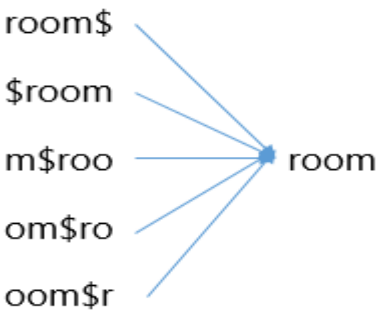
برای کوئری rom* عبارت $r and ro and om جستجو می‌شود و حاصل جستجو :

romantic , room









برای کوئری *rom* عبارت rom* جستجو می‌شود و حاصل جستجو :

chrome , from , romantic

برای کوئری rom* عبارت $rom* جستجو می‌شود و حاصل جستجو :

romantic

نتیجه :

در کوئری *rom* جواب‌های room , agronomy و در کوئری rom* جواب room صحیح نیستند ولی در خروجی bigram index دیده می‌شود. بنابراین همانطور که مشاهده می‌شود در هر دو کوئری عملکرد روش permuterm index بهتر از bigram index است.

۲- در یک موتور جست و جو کلمه "herq" جست و جو شده است ،اما این کلمه در دیکشنری این موتور جست و جو یافت نمی شود . با توجه به پارامترهای داده شده(تعداد رخداد حروف، تعداد رخداد خطاها و احتمال رخداد کلمات مختلف)، بنظر شما کلمه‌ی وارد شده به اشتباه به جای کدام یک از کلمات زیر وارد شده است ؟

پاسخ:

herq : { “here”, ”hero”, ”her”}

تعداد تکرار حرف e : C(e) =593086170

تعداد تکرار حرف r : C(r) = 32534251

تعداد تکرار حرف o: C(o) = 246429812

C(q|e) = 1

C(q|o) = 1

C(rq|r) = 1

P(“here”) = 0.000624

P(“hero”) = 0.00001204

P(“her”) = 0.00038

Candidate word	correct letter	Error letter	X|W	P(X|W)	P(W)	P(X|W) P(W)*10^{10}
Here	e	q	q|e	1/593086170 =0.00000000016	0.000624	0.009
Her	-	q	rq|r	1/32534251 = 0.0000000308	0.00038	0.1
Hero	o	q	q|o	1/246429812 =0.00000000405	0.00001204	0.0004

با توجه به احتمالات بدست آمده کلمه‌ی "her" انتخاب می‌شود .

فصل پنجم : compression

۳-نشان دهید که اندازه لغت نامه در قانون Zipf محدود و در قانون Heaps نامحدود است.

پاسخ :

در قانون Heaps اندازه لغت‌نامه همراه با افزایش تعداد توکن‌ها افزایش می‌یابد، بنابراین می‌تواند تا بی‌نهایت پیش برود.

اما در قانون Zipf اندازه لغت‌نامه باید در ابتدا یک عدد ثابت در نظر گرفته شود، پس محدود است.

۴- الف) در یک سیستم بازیابی اطلاعات یک دیکشنری با ۴۰۰۰۰۰ کلمه داریم، میانگین طول هر کلمه هشت کاراکتر است. اگر بخواهیم این دیکشنری را با روش بلوک بندی و k=8 ذخیره کنیم ،د ر حالی که هر اشاره گر به postings list ،۴ بایت و تعداد تکرار هر کلمه (term frequency) در ۴ بایت ذخیره شود و برای اشاره به ابتدای هر بلوک از یک اشاره گر ۳ بایتی استفاده شود ،با این شرایط حافظه مورد نیاز را برای ذخیره این دیکشنری بدست آورید .

در این حالت، نسبت به حالت عادی و بدون هیچ گونه فشرده‌سازی (تخصیص ۲۰ بایت برای ذخیره‌سازی هر کلمه) چند درصد کاهش حجم حافظه خواهیم داشت ؟

ب) اگر یک دیکشنری شامل مجموعه کلمه زیر داشته باشیم ،میانگین تعداد مقایسه‌هایی را که باید انجام شود تا یک کلمه را پیدا کنیم در دو حالت زیر محاسبه کنید:

۱.بدون فشرده سازی دیکشنری

۲.با فشرده سازی به روش بلوک بندی با k=4

{آباد، انقلاب، پیشرفت، تورم، دانشگاه، درخت، رمضان، سلامت، سیاست، فقر، کرونا، وطن}

پاسخ :

الف)

۴۰۰۰۰۰ کلمه با k = 8 ،به۵۰۰۰۰ بلوک تقسیم می‌شود .

۱. اگر هر کلمه ۸ کاراکتر باشد و یک کاراکتر برای نگه داری طول آن ،هر کلمه ۹ بایت حافظه نیاز دارد : ۹*۴۰۰۰۰۰
۲. هر بلوک به یک اشاره گر ۳ بایتی نیاز دارد : ۳*۵۰۰۰۰
۳. ۴ بایت برای تعداد تکرار کلمات و ۴ بایت برای اشاره گر به postings list : ۸*۴۰۰۰۰۰

.۴ .1+.2+.3=6950000Byte =6.95MB

بدون فشرده سازی :

۱. ۴ بایت برای تعداد تکرار کلمات و ۴ بایت برای اشاره گر به postings list : ۸*۴۰۰۰۰۰
۲. ۲۰ بایت برای هر کلمه : ۲۰*۴۰۰۰۰۰

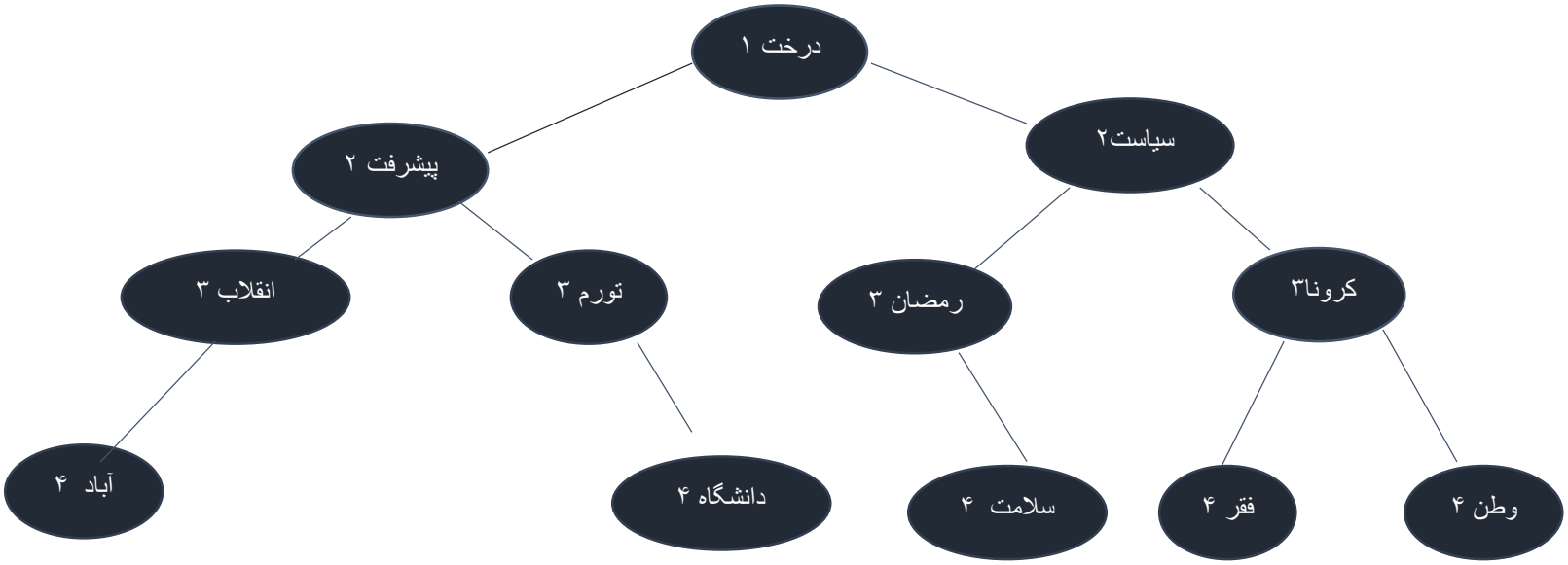
.۳ .1+.2=11600000Byte =11.6MB

**حافظه مورد نیاز ۴۰ درصد کاهش یافته است.

ب)

{آباد، انقلاب، پیشرفت، تورم، دانشگاه، درخت، رمضان، سلامت، سیاست، فقر، کرونا، وطن}

در حالتی که از روش‌های فشرده سازی استفاده نمی کنیم ،جست و جو بر اساس درخت دودویی زیر انجام می‌شود .



** تعداد جست و جوی مورد نیاز برای هر کلمه در مقال آن نوشته شده است .

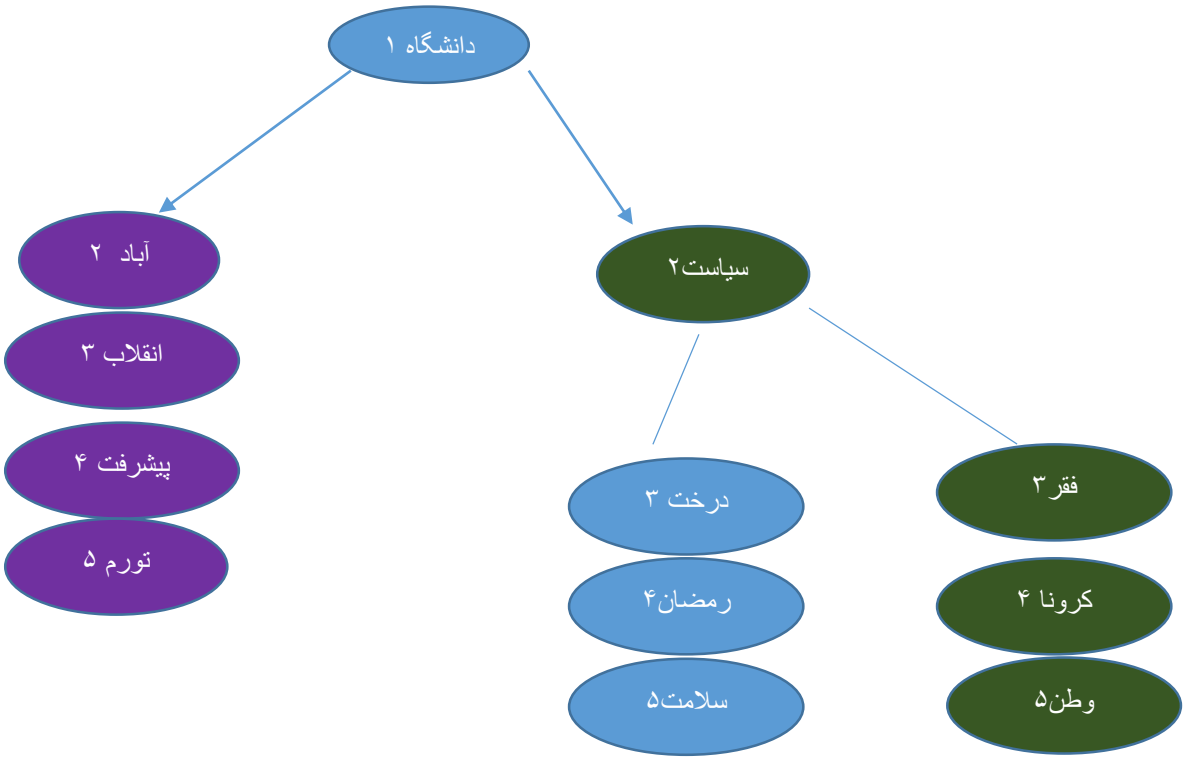
(1+2*2+4*3+4*4)/12 = 3.08

در حالتی که از روش بلوک بندی استفاده می کنیم :

داده‌ها به سه بلوک چهارتایی تقسیم می‌شوند. ابتدا یک جستجوی دودویی بین سه ریشه درخت انجام می شود تا بلوک مورد نظر پیدا شود و وقتی که بلوک پیدا شد داده های بلوک به صورت ترتیبی پیمایش می شوند.

کلمه ورودی باکلمه دانشگاه مقایسه می شود. اگر برابر بود که خاتمه می یابد. اگر کوچکتر بود که لیست سمت چپ به ترتیب بررسی می شود .

اگر بزرگتر بود می بایست یک مقایسه با کلمه سیاست انجام شود. اگر برابر بود که خاتمه می یابد. اگر بزرگتر بود که بلوک سمت راست به ترتیب بررسی می شود و اگر کوچکتر بود که بلوک وسط به ترتیب بررسی می شود.



(1*1+2*2+3*3+4*3+5*3)/12 = 3.41

۵- یک لغت نامه ی شاخص معکوس شامل کلمات زیر است :

elite ,elope ,ellipse ,eloquent ,eligible ,elongate

الف) فرض کنید دیکشنری در این شاخص معکوس ابتدا کلمات را مرتب و به صورت یک رشته‌ی طولانی ذخیره می‌کند. اگر از روش فشرده‌سازی front coding استفاده شود، رشته‌ی ذخیره شده به ازای بلوک با سایز های ۳ و ۶ را بنویسید.

پاسخ :

بلوک سه تایی : 8el*igible3◇ite5◇lipse8elo*ngate2◇pe5◇quent

بلوک ۶ تایی : 8el*igible3◇ite5◇lipse6◇ongate3◇ope6◇oquent

ب) فرض کنید :

- در دیکشنری کلمه، تعداد تکرار کلمه و یک اشاره‌گر به postings list ذخیره می‌شود.
- تعداد تکرار کلمه و اشاره‌گر به postings list هر کدام ۴ بایت فضا لازم دارند.
- هر کاراکتر در یک بایت ذخیره می‌شود.
- برای ذخیره سازی به روش بلوک‌بندی، هر اشاره‌گر به کلمه، ۳ بایت فضا اشغال می‌کند.
- هر کلمه به صورت میانگین ۸ کاراکتر دارد.
- ذخیره سازی با بلوک‌های ۳تایی و استفاده از فشرده‌سازی front coding، باعث کاهش ۲۰% از طول رشته ی ذخیره شده می‌شود.

با توجه به فرضیات ذکر شده سایز بزرگترین لغت نامه ای که می‌توانیم ذخیره کنیم چقدر است؟ حافظه‌ی لازم برای ذخیره کردن بزرگترین دیکشنری ممکن چقدر است؟

پاسخ :

پس طول رشته‌ی کد شده حداکثر می‌تواند 2^{24} بیت باشد. → 3 byte = 24 bit

طول رشته‌ی واقعی : $1.25 * 2^{24} = 20\ M$ → 20 MB = V * 8B → V = 2500000 کلمه

حافظه‌ی مصرفی هر بلوک : اشاره‌گر به کلمه + (تعداد تکرار کلمه + اشاره‌گر به لیست پستینگ) * سایز بلوک

3*(4 + 4) + 3 = 27 byte

تعداد بلوک‌های لازم برای ۲۵۰۰۰۰۰ کلمه : 2500000 / 3 = 833334

حافظه مصرفی کل : 833334 * 27 + 16M = 38.5 MB (16M حافظه مصرفی برای ذخیره سازی رشته‌ی کد شده است.)

۶-الف) کلمه "ایران" در دیکشنری یک سیستم بازیابی اطلاعات در سندهایی با شماره های زیر آمده است ،این اطلاعات را به روش کدگذاری گاما فشرده کنید .

۲۰, ۵۲,۷۸,۱۵۶

ب)اگرpostings list را به روش کد گاما ذخیره کنیم ،میزان حافظه مورد نیاز برای هر عضو postings list چه قدر است ؟

پاسخ :

الف)

docIds :۲۰, ۵۲,۷۸,۱۵۶ ⟹ gaps : ۲۰,۳۲,۲۶, ۷۸

⟹ 10100,100000,11010,1001110

number	length	Offset	Gamma code
20	11110	0100	11110,0100
32	111110	00000	111110,00000
26	11110	1010	11110,1010
78	1111110	001110	1111110,001110

ب)

اگر عدد دهدهی Gرا به مبنای دو تبدیل کنیم $\lfloor log_2^G \rfloor + 1$ رقم خواهد داشت .

در قسمت length ، به طول معادل دودیی G منهای رقم اول آن، یک و به همراه یک صفر در انتهای آن نگه داری میشود ،پس طول قسمت length برابر با $\lfloor log_2^G \rfloor + 1$ است .

در قسمت offset ، عدد دودیی معادل G با حذف پرارزش ترین رقم آن نگه‌داری می‌شود ،پس طول آن $\lfloor log_2^G \rfloor$ خواهد بود .

در نتیجه برای عدد G ، $\lfloor log_2^G \rfloor * 2 + 1$ بیت حافظه نیاز است .

۷- در ادامه قسمتی از یک دیکشنری ارائه شده است که postings list آن به روش VB ذخیره شده است ، با توجه به این دیکشنری برای پرسمان " تغذیه سالم " چه نتیجه‌ایی به کاربر برگردانده می شود ؟

پاسخ :

ابتدا شماره متون را به دهدهی تبدیل میکنیم،میدانیم در این حالت عدد اول شماره متن اول (docId)و شماره های دیگر هر کدام فاصله از شماره متن قبلی (gap)هستند:

تغذیه	10000000(0)-10001100(12)-0000000110001010(138)-10100100(36)
روان	10000001(1)-10000111(7)-11101010(106)
سالم	10001010(10)-10000010(2)-)-0000000110001010(138)-10101000(40)
علم	11000011(67)-10011000(24)-10100011(35)

تغذیه : 0,12,150,186

روان : 1,8,114

سالم : 10,12,150,190

علم : 67,91,26

بنابراین پاسخ پرسمان "تغذیه سالم " متن هایی با شماره ۱۲ و ۱۵۰ است .