6.اقرأ الرقم الثالث و سمه C

7.S = S + C (اضافة الرقم الثالث C الى العداد ليخزن في S)

8.اطبع S (بذلك يكون المتغير S قد احتفظ بكل القيم السابقة مجتمعة)

اذا نفذت الخوارزمية على سلسلة الأعداد التالية 1، 5، 10 سيكون التغير في الذاكرة كما هو موضح في الجدول (17-4).

المتغير	S	A	B	C
+	0	-1	5	10
S	0	1	6	16

جدول (17-4): التغير في الذاكرة.

مثال:

اكتب خوارزمية لايجاد معدل ثلاثة أرقام.

خوارزمية هذا المثال تشبه تماماً خوارزمية المثال السابق مع إضافة أمر واحد يقوم بقسمة مجموع الأعداد على ثلاثة. الخوارزمية التي سنكتبها الآن ستقلص عدد المتغيرات التي نقرأ بها الأرقام الى متغير واحد. إن تقليص عدد المتغيرات يعني حجز مكان أقل في الذاكرة، وهذا أفضل من الناحية العملية.

S = 0.1

2.اقرا العدد الاول وسمه A

S = S + A.3

4.اقرا العدد الثاني وسمه A

S = S + A.5

6.اقرا العدد الثالث وسمه A

S = S + A.7

Av = S / 3.8

9.اطبع Av

اذا نفذت الخوارزمية على سلسلة الأعداد التالية 1، 4، 10 سيكون التغير في الذاكرة كما موضح في الجدول (17-5).....

S	0	1	5	15
A	1	4	10	
Av	S/3			5

الجدول (17-5): التغير في الذاكرة.

17-3-5 التكرار

ان التكرار واحد من أهم المفاهيم في مجال البرمجة وسنوضحه جيداً بواسطة الأمثلة.

مثال:

اكتب الخوارزمية لايجاد معدل 100 عدد.

خوارزمية هذا المثال تشبه تماماً خوارزمية المثال السابق ما عدا ان عدد الأرقام هو 100 بدلاً من ثلاثة، فيما يلي الخطوات الثلاث الأولى من الخوارزمية:

S = 0.1

2.اقرأ العدد التالي وسمه A

S = S + A.3

وسيتبادر إلى الذهن أن الخطوتين الرابعة والخامسة ستكونان

4.اقرأ العدد التالي وسمه A

S = S + A.5

سيتبادر إلى الذهن أن الخطوتين السادسة والسابعة ستكونان

6.اقرأ العدد التالي وسمه A

S = S + A.7

وهكذا سيتم تكرار هاتين الخطوتين لكل عدد مرة، وهذا من غير المعقول في مجال البرمجة وكتابة الخوارزميات، فتصور لو طلب كتابة خوارزمية لحساب معدل أعمار مليون انسان، سيصبح من المستحيل كتابة خوارزمية بهذا الأسلوب. هنا تأتي أهمية التكرار وهو أسلوب يستخدم لمثل هذه الحالات لتكرار عدد من الأوامر عدة مرات. باستخدام أسلوب التكرار يمكن كتابة الخوارزمية بالشكل التالي:

S = 0.1

2.كرر ما يأتي 100 مرة

3.اقرا العدد التالي وسمه A

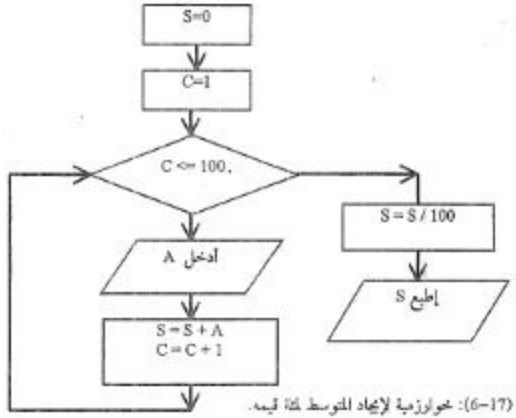
S = S + A.4

5.نهاية التكرار

S = S / 100.6

7.اطبع S

ويمكننا رسم الخوارزمية كما في الشكل (17-6).



الشكل (17-6): خوارزمية لإيجاد المتوسط لمئة قيمه.

17-4 البرمجة الهيكلية

لقد احدث مبدأ البرمجة الهيكلية (Structured Programming) والذي استحدثت أجيال تغييراً جذرياً في اساليب البرمجة القديمة الموروثة من لغة فورتران، وقد استخدمت لغة فورتران القديمة في اسلوب البرمجة وطريقة التفكير جملة القفز (GO TO) والتي اعتبرت من اكبر عيوب فورتران. والسبب الرئيس لنبذ جملة القفز هو انها تؤدي الى صعوبة في البرمجة وصعوبة في تتبع خطوات البرنامج وفهمه خاصة من شخص غير الشخص الذي كتبه. وفي السبعينات من القرن الماضي اشتهر الاسم الإصطلاحي (Sphagetti Program) أي برنامج المكرونة، والذي اطلق على اي برنامج يحتوي على اكثر من جملة قفز واحدة. وسبب هذه التسمية هو كثرة التشابك والتشعب في البرنامج الذي يكثر من استخدام جملة القفز GO TO، مما يؤدي الى صعوبة المتابعة.

يتلخص أسلوب البرمجة الهيكلية بانه يمكن حل اي مسألة باستخدام ثلاثة انواع من الجمل، وهي:

1.جمل التتابع (Sequential Statements).

2.جمل الاختيار (Selection Statements).

3.جمل التكرار (Repetition Statements).

ولا يسمح في البرمجة الهيكلية باستخدام جملة القفز بتاتاً، مع العلم انها متوفرة في معظم اللغات الحديثة والتي صممت اعتماداً على أسلوب البرمجة الهيكلية.

ويقصد بالتتابع ان الجمل تنفذ بالتتابع جملة جملة بنفس الترتيب الذي وضعت فيه ولا تنفذ الجملة إلا مرة واحدة. وكما هو موضح في الشكل (17-7)، فان أ ستنفذ قبل ب وستنفذ كل منهما مرة واحدة. وخطوات الخوارزمية في المثال السابق تمثل هذا النوع من الجمل حيث انها جميعاً خطوات تتابع.