

## خوارزمية تحويل صيغة (infix) الى (postfix)

1- نستخدم مكديسين ، المكديس الاول (ST1) لخزن المتغيرات (العوامل (operands) وفي الخطوة الاخيرة ستتجمع فيه الصيغة النهائية (صيغة Postfix) والمكديس الثاني (ST2) يستخدم لخزن اشارات العمليات الحسابية (Operators).

- 2- نفحص التعبير الحسابي رمزا رمزا من اليسار الى اليمين .  
3- عند كل رمز نقوم بما يأتي :-

ينفذ ما يأتي :	اذا كان الرمز :
+ يخزن (push) في المكديس (ST1)	+ احد العوامل (operand)
+ يخزن (push) في المكديس (ST2)	+ قوس ايسر
+ اخراج (pop) جميع الرموز من المكديس (ST1) وخزنها (push) بالتتابع في المكديس (ST2) لغاية الوصول الى القوس الايسر الذي يجب اخراجه واهماله مع القوس الايمن .	+ قوس ايمن
+ اخراج (pop) جميع العمليات الحسابية (ان وجدت) في المكديس (ST2) التي اسبقيتها اعلى او تساوي اسبقية العملية الحسابية الحالية وخزنها في المكديس (ST1) (التوقف عن ذلك عند عدم تحقق الشرط) ومن ثم خزن العملية الجديدة في المكديس (ST2).	+ عملية حسابية (operator)
4- عند انتهاء كل رموز التعبير الحسابي يتم اخراج (pop) جميع الرموز المتبقية في المكديس (ST2) بالتتابع وخزنها (push) في المكديس (ST1) الذي يحوي الصيغة النهائية (postfix).	

مثال : حول العبارة الحسابية التالية من صيغة (infix) الى صيغة (postfix)  
 $a - b * (c + d) / (e - f) ^ g * h$

المكدس الثاني ST2	المكدس الاول ST1	الرمز المدخل	الحل : رقم الخطوة
.....	a	a	1
-	a	-	2
-	ab	b	3
- *	ab	*	4
- * (	ab	(	5
- * (	abc	c	6
- * ( +	abc	+	7
- * ( +	abcd	d	8
- *	abcd +	)	9

نلاحظ هنا عند ورود القوس الايمن يتم اخراج (نقل) جميع العمليات الحسابية لغاية القوس الايسر من المكدس (ST2) الى (ST1) مع اخراج القوس الايسر ليكمل هو والقوس الايمن .

- /	abcd + *	/	10
- / (	abcd + *	(	11
- / (	abcd + * e	e	12
- / ( -	abcd + * e	-	13
- / ( -	abcd + * ef	f	14
- / )	abcd + * ef-	)	15
- / ^	abcd + * ef-	^	16
- / ^	abcd + * ef-g	g	17
- * /	abcd + * ef-g ^ /	*	18

لان اسبقية الضرب (\*) = > اسبقية الرفع (^) والقسمة (/)

- *	abcd + * ef-g ^ /h	h	19
-----	--------------------	---	----

هنا انتهت جميع المدخلات لذا ينقل المتبقي في المكدس

(ST2) الى المكدس (ST1) بالتتابع ليصبح

.....	abcd + * ef-g ^ /h* -	-	20
-------	-----------------------	---	----