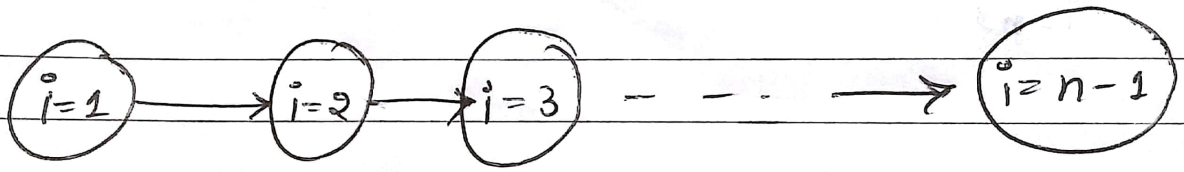
Question 3:




Width = 1
Work = n-1
CP = n-1.

Question 3.1.2:

```
template <Typename T, typename op>
T reduce (T* array , size_t n)
{ int res = 0;
while (n>1)
{
int j=0;
for (int i=0; i<=n-1; i+=2)
{
if (i!=n-1)
{
arr[j] = op(arr[i], arr[i-1]);
}
else
arr[j] = arr[0];
j++;
}
reduce ( *array, n/2);
}
res = arr[0];
return res;
}
```

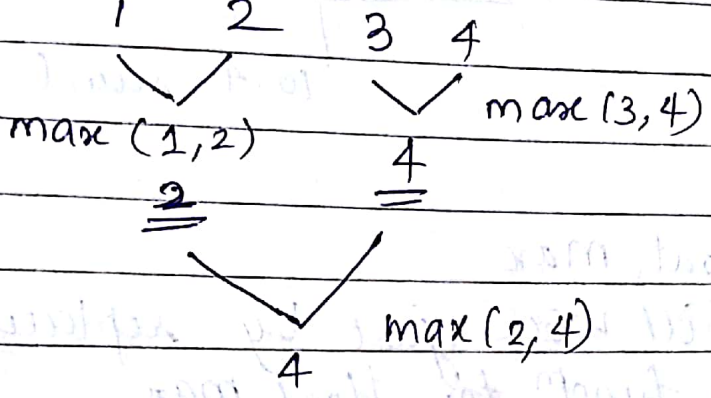
3.2

① Yes, the parallel version will work for max too. by substituting op by max()

For max:

eg:-

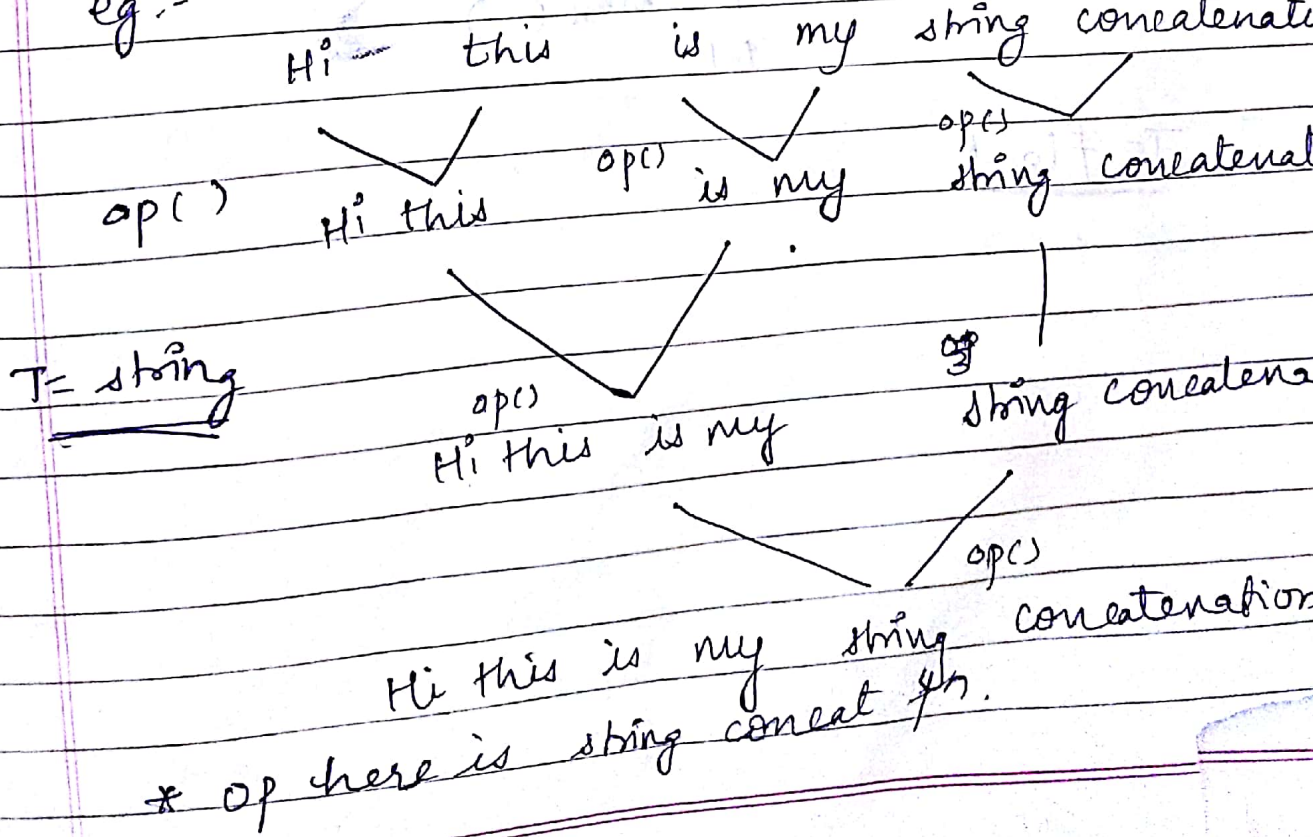
T = int




② for string concat :-

Yes, the code can also be used for achieving parallelism in concatenation. by substituting op() by string concat()

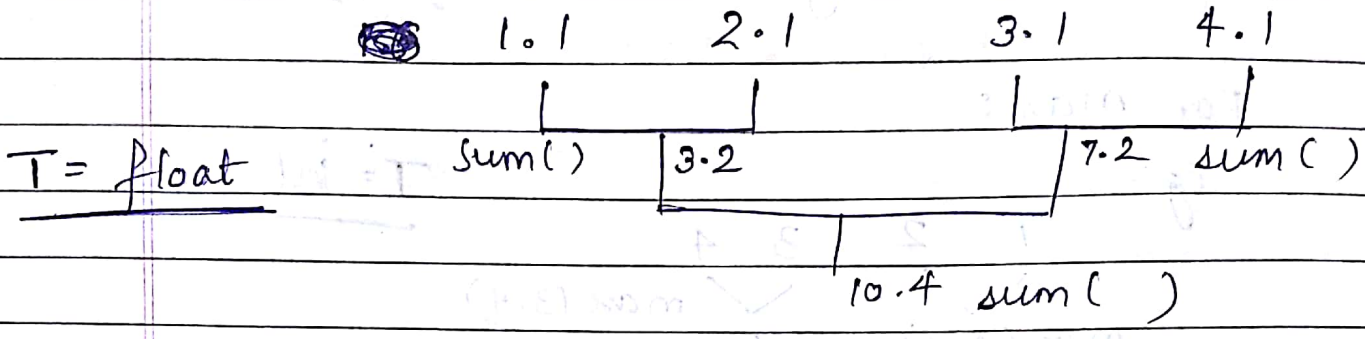
eg:-




* op there is string concat fn.

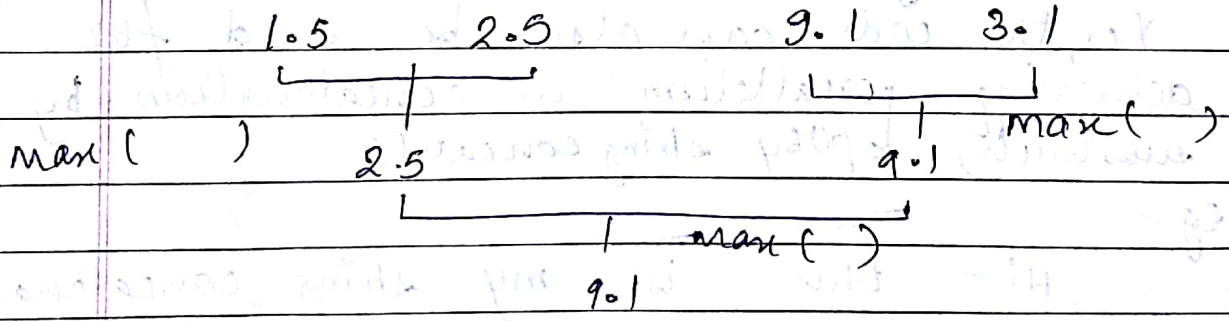
Yes, It will work just by replacing op by float sum funcn.

3) for float, sum.




4) for float, max

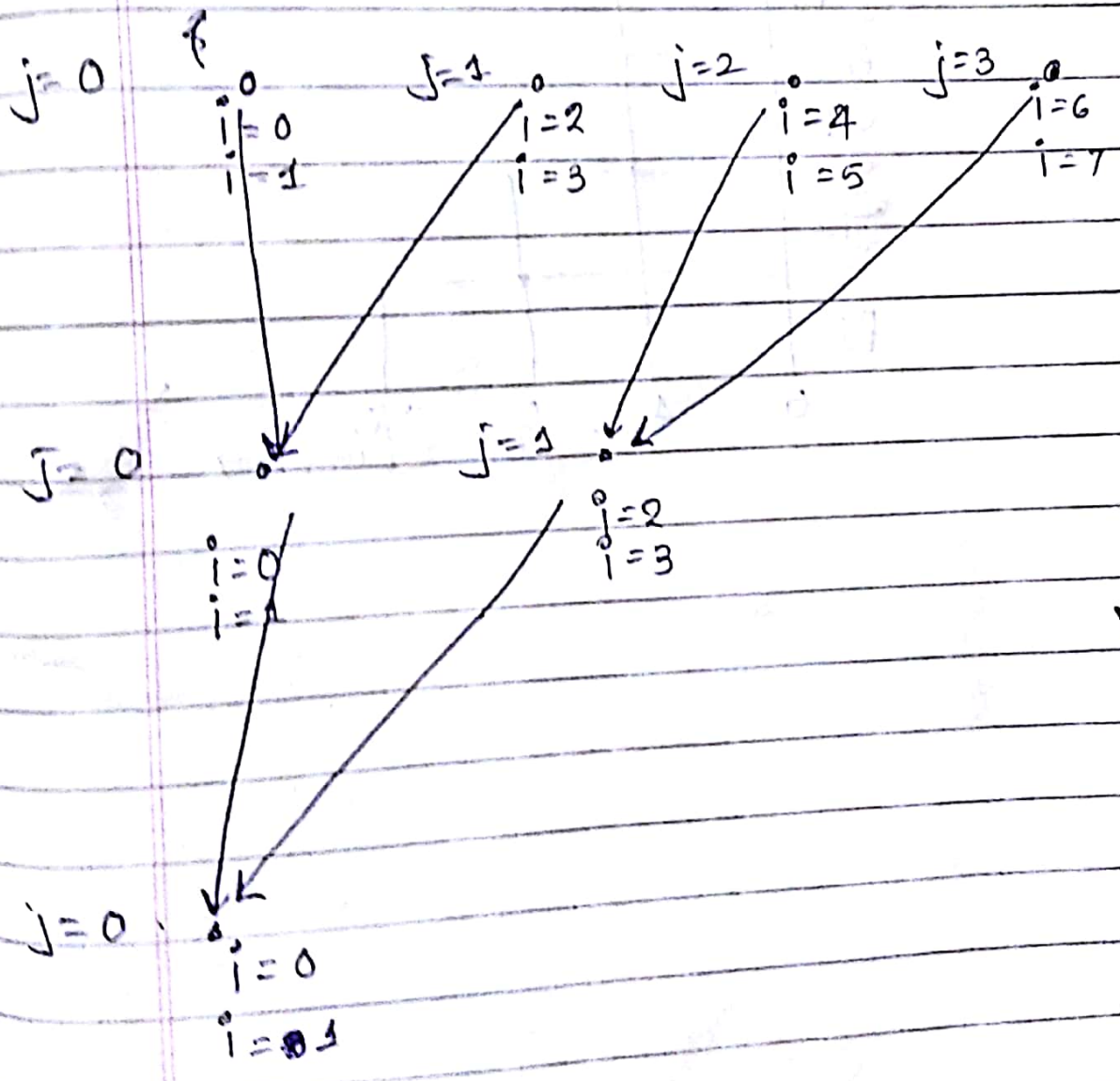
Yes, it will work just by replacing op by max funcn for float max




T= float

Q: 3.1

Consider $n = 8$




width = $n/2$

Work = $(n-1)$

CP = $n/4$

i = 1
j = 1

Questn 3: