S → AC | SB
A → bASC | a
B → aSB | BbC
C → Bc | ad

→ here S → AC | SB } are useful while
A → bASC | a
C → Bc | ad
B → aSB | BbC → useless,

So, S → AC
A → bASC | a
C → ad

A → xyz | YzyzZ
X → Xz | xYx
Y → yYy | XZ
Z → Zy | z

→ here X → Xz | xYx and
Y → yYy | XZ are useless

So remove X, Y

$A \rightarrow xyz$
$z \rightarrow zy \mid z$

here $z \rightarrow zy \mid z$ is also useless cz

$A \rightarrow xyz$ these is not a Non terminating z

So $A \rightarrow xyz$

$S \rightarrow AB \mid CA$
$B \rightarrow BC \mid AB$
$A \rightarrow a$
$C \rightarrow aB \mid b$

here B is useless cz these is not a terminating symbol,

$S \rightarrow CA$
$A \rightarrow a$
$C \rightarrow b$

$S \to ABCd \mid ABd \mid ACd \mid BCd \mid Ad \mid Bd \mid Cd \mid d$

$A \to BC \mid B \mid C$
$B \to bB \mid b$
$C \to cC \mid c$

$\Rightarrow S \to ABCd \mid ABd \mid ACd \mid BCd \mid Ad \mid Bd \mid Cd \mid d$

$A \to BC \mid b \mid c \mid bB \mid cC$
$B \to bB \mid b$
$C \to cC \mid c$

Remove Null

$S \to AaA$
$A \to Sb \mid bCC \mid \epsilon$
$C \to CC \mid abb$

$\Rightarrow S \to AaA \mid aA \mid Aa \mid a$
$A \to Sb \mid bCC$

$C \to CC \mid abb$

#

$S \rightarrow aAB \mid dA$

$A \rightarrow bAc \mid \epsilon$

$B \rightarrow dB \mid \epsilon$

$\Rightarrow$

$S \rightarrow aAB \mid dA \mid aB \mid d \mid aA \mid a$

$A \rightarrow bAC \mid bc$

$B \rightarrow dB \mid d$

#

$S \rightarrow ABC$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

$C \rightarrow C$

$\Rightarrow$

$S \rightarrow ABC \mid BC \mid AC \mid C$

$A \rightarrow aA \mid a$

$B \rightarrow bA \mid b$

$C \rightarrow \epsilon$

#

$S \rightarrow aBDh$

$B \rightarrow Bb \mid c$

$D \rightarrow EF$

$E \rightarrow g \mid \epsilon$

$F \rightarrow f \mid \epsilon$

$S \rightarrow aBDh$

$B \rightarrow cB'$

$B' \rightarrow bB' \mid \epsilon$

$D \rightarrow EF$

$E \rightarrow g \mid \epsilon$

$F \rightarrow f \mid \epsilon$

#

$A \rightarrow Ab \mid Aab \mid bA \mid a$

$bA, a \rightarrow B$

$Ab; Aab \rightarrow d$

$A \rightarrow aA' \mid bAA'$

$A' \rightarrow bA' \mid \epsilon \mid [illegible], AbA'$

S → ABC
A → Aa | d
B → Bb | e
C → Cc | f

=> S → ABC
A → dA'
A' → aA' | ∈
B → eB'
B' → bB' | ∈
C → fC' |
C' → cC' | ∈

# int a[10], b[10], dot_Prod, i;
dot_Prod = 0;
for (i=0; i<10; i++) dot_Prod += a[i] * b[i]

=>

1. dot_Prod := 0;
2. i := 0
3. if i < 10 goto (15)
4. goto (5)
5. $t_1 = i$
6. $t_2 = 4$
7. $t_3 = t_2 * t_1$
8. $t_4 = a + t_3$
9. $t_5 = b + t_3$
10. $t_6 = (*t_5) * (*t_4)$
11. $t_7 = dot\_Prod + t_6$
12. $dot\_Prod = t_7$
13. $i = i + 1$
14. goto (3)
15. ——

```
# int a[10], b[10], dot_Prod, i;
  int* a1; int* b1;
  dot_Prod = 0;
  a1 = a;
  b1 = b;
  for (i = 0; i < 10; i++)
      dot_Prod += *a1++ * *b1++
```

```
dot_prod = 0;
a1 = &a;
b1 = &b;
i = 0
L1: if i >= 10 goto (L2)
    T3 = *a1
    T4 = a1 + 1
    a1 = T4
    T5 = *b1
    T6 = b1 + 1
    b1 = T6
    T7 = T3 * T5
    T8 = dot_prod + T7
    dot_prod = T8
    T9 = i + 1
    i = T9
    goto L1
L2:
```

#

```
x = 1;
y = x + 10;
while (x < y) {
    x = x + 1;
    if (x % 2 == 1) then y = y + 2;
    else y = y - 2;
}
```

```
x = 1;
T1 = x + 10
y = T1
L1: if x >= y goto (L2)
T2 = x + 1
x = T2
T3 = x % 2
L3: if x > 1 goto (L4)
T4 = y + 1
y = T4 goto (L1)
L4: T5 = y - 2
y = T5
goto (L1)
L2:
```

#

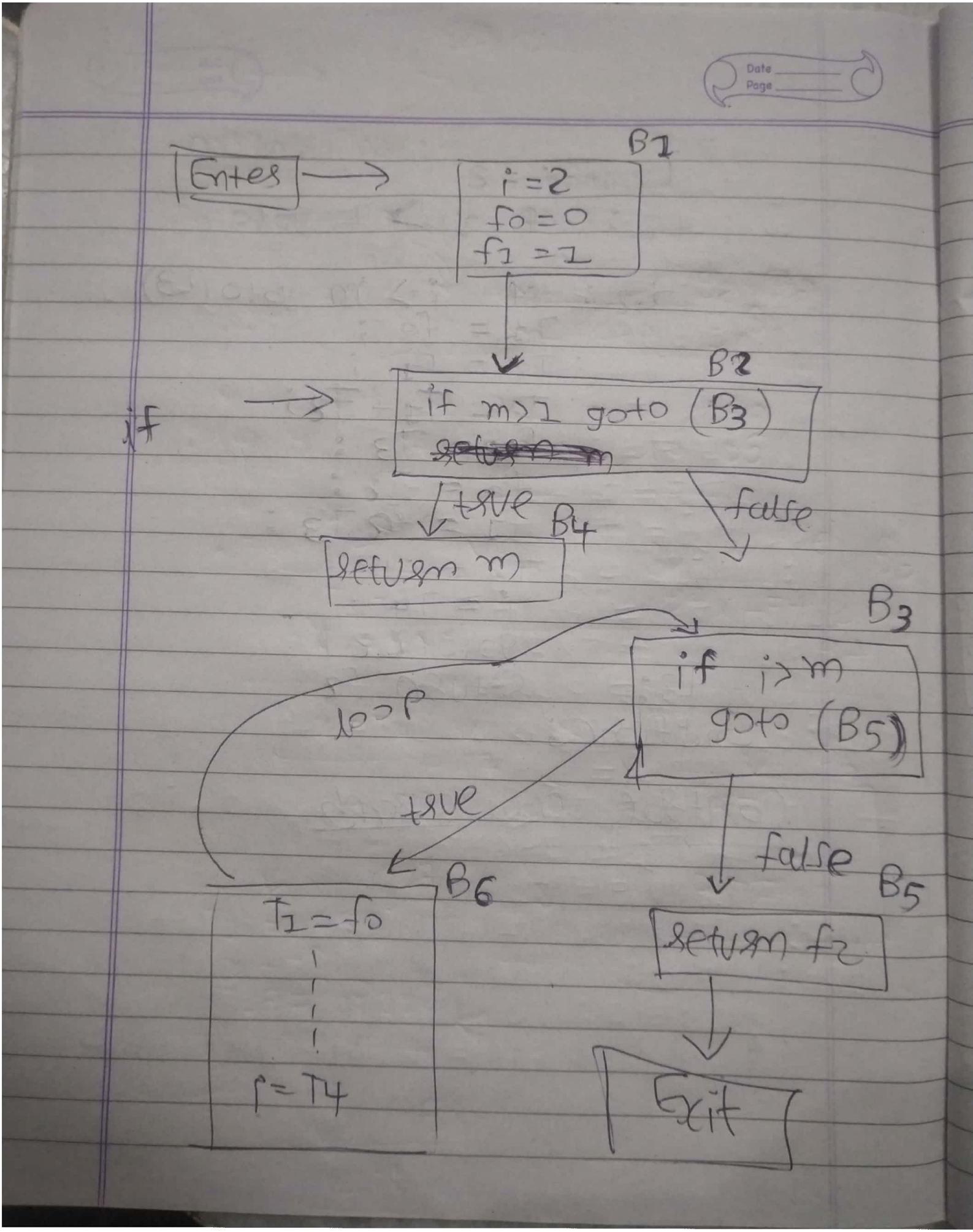
```
Unsigned int fib(m)
{
    Unsigned int m;
{
    Unsigned int fo=0, f1=1, f2, i;
    if (m<=1) {
            return m;
    }
    else {
        for (i=2; i<=m ; i++){
            fo = fo+f1;
            fo = f1;
            f1 = f2;
        }
        return f2;
    }
}

fib:
    Start func;
        i = 2
        fo = 0 ;
        f1 = 1 ;
L1:  if m > 1  goto (L2)
        return m
```

~~L1: i = 2~~

~~L2: if i > m goto L3~~

```
L2: if  i > m goto (L3)
    T1 = f0 ;
    T2 = f1 ;
    T3 = T1 + T2 ;
    f2 = T3 ;
    f0 = T2 ;
    f1 = T3 ;
    T4 = i + 2
    i = T4
    goto (L2)
L3: return f2
EndFunc ;
```

Control flow graph



int A[5], x, i, n;

```
for ( i=1 ; i<=n; i++) {
    if ( i<n ) {
        x = A[i];
    }
    else {
        while ( x>4) {
            x = x*2 + A[i];
        };
    };
x = x + 5;
}
```

3AC :-

```
    i = 1
L1: if i>n goto (L2)
    if i>=n goto (L3)
    T1 = i
    T2 = 4 * i
    T3 = A + T2
    T4 = *(T3)
    x = T4
    goto (L4)
L3: if x<=4 goto (L4)
    T5 = i
    T6 = 4
    T7 = T5 * T6
    T8 = A + T7
    T9 = *(T8)
    T10 = x * 2
    T11 = T10 + T9
    x = T11
    goto (L3)
L4: T12 = x + 5
    x = T12
    T13 = i + 1
    i = T13
    goto (L1)
L2:
```

Control flow:-