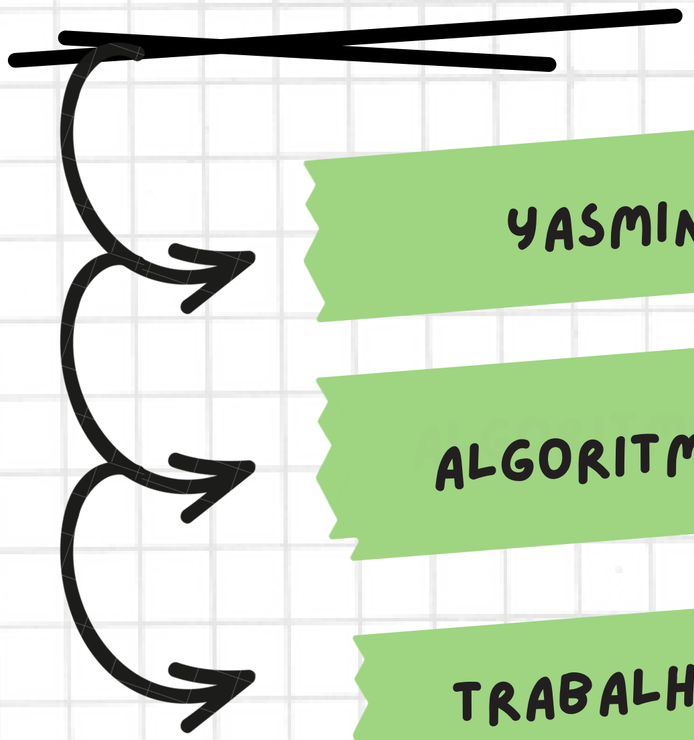
LISTAS



YASMIN

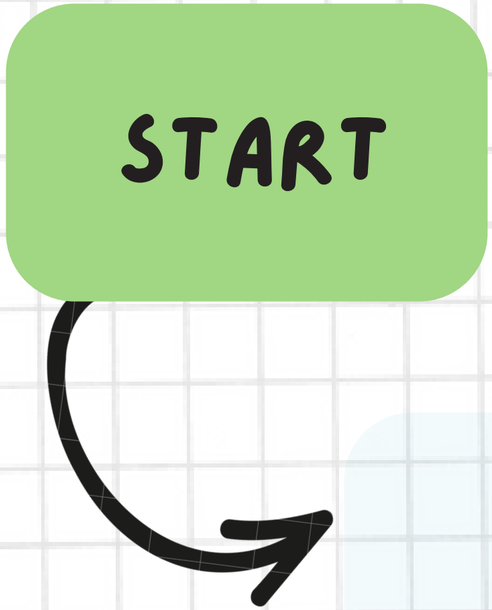
ALGORITMOS E ESTRUTURAS DE DADOS

TRABALHO EXTRA

```
TYPEDEF STRUCT _LISTA {
     CHAR *NOME;
     STRUCT _LISTA *NEXT;
} LISTA;
```

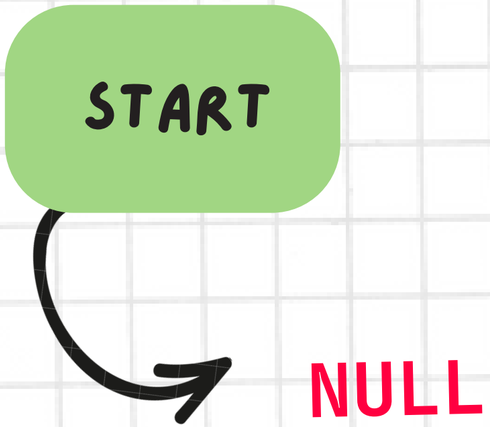
```
TYPEDEF STRUCT _LISTA {
    CHAR *NOME;
    STRUCT _LISTA *NEXT;
} LISTA;
```

```c
TYPEDEF STRUCT _LISTA {
    CHAR *NOME;
    STRUCT _LISTA *NEXT;
} LISTA;
```



NULL

```
TYPEDEF STRUCT _PESSOA {
    CHAR *NOME;
    STRUCT _PESSOA *NEXT;
} PESSOA;

LISTA *START = NULL;
```

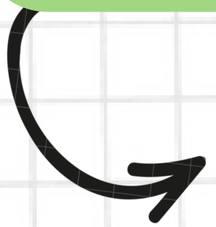

START
NULL

NEWELEMENT("MARIA")

NULL

NEWELEMENT("MARIA")

```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```



NULL

NEWELEMENT("MARIA")

```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```

NULL

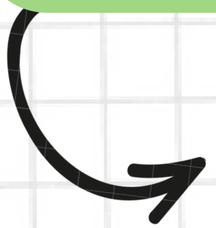
NEWELEMENT("MARIA")

```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```

NULL

NEWELEMENT("MARIA")

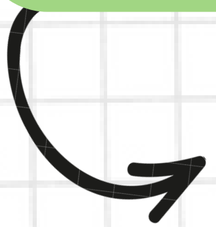
```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```



NULL

NEWELEMENT("MARIA")

```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```



NULL

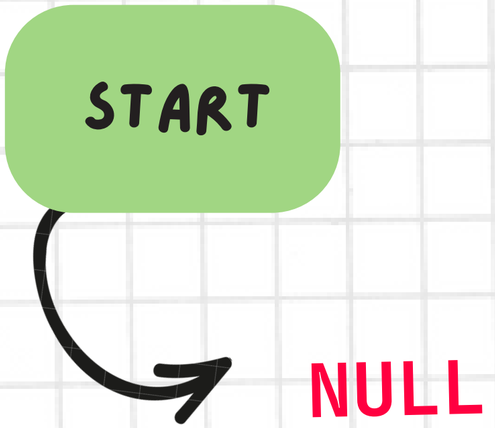
NEWELEMENT("MARIA")

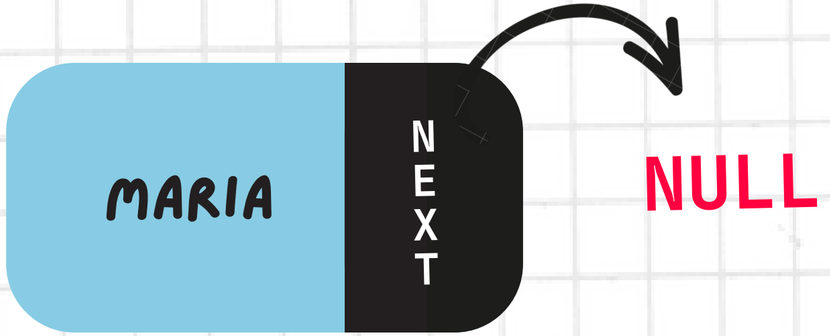
```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```

NULL

NEWELEMENT("MARIA")

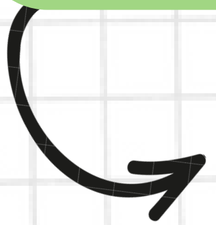
```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```



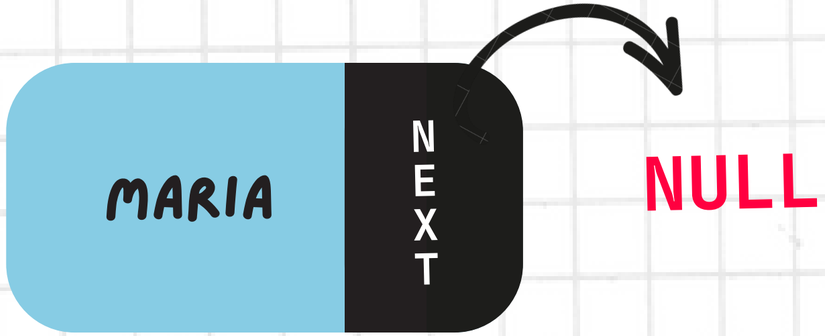



```
NEWELEMENT("MARIA")
```

```
PESSOA *NEWELEMENT(CHAR *TEXT)
{
    PESSOA *NEWP;
    NEWP = (PESSOA *) MALLOC (SIZEOF(PESSOA));
    NEWP -> NOME = (CHAR *)MALLOC(STRLEN(TEXT) + 1);
    STRCPY(NEWP -> NOME, TEXT);
    NEWP -> NEXT = NULL;
    RETURN NEWP;
}
```

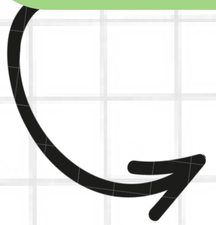


NULL

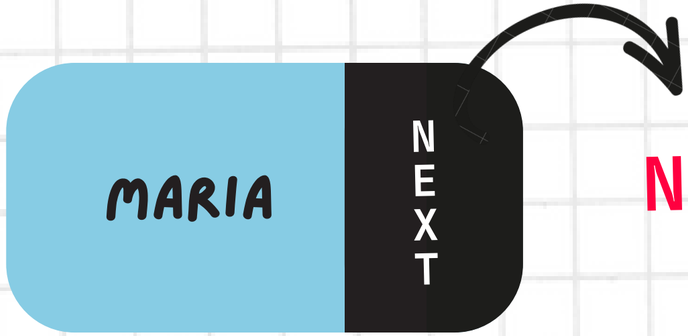


NULL

NEWELEMENT("MARIA")

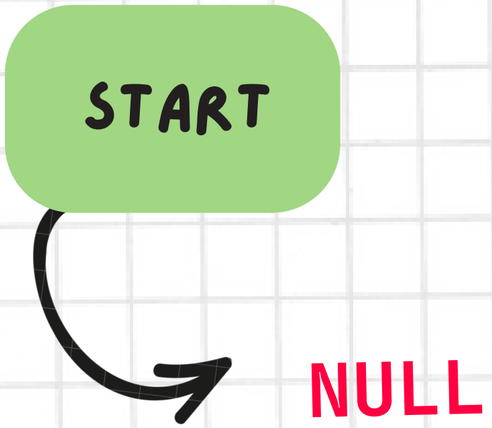


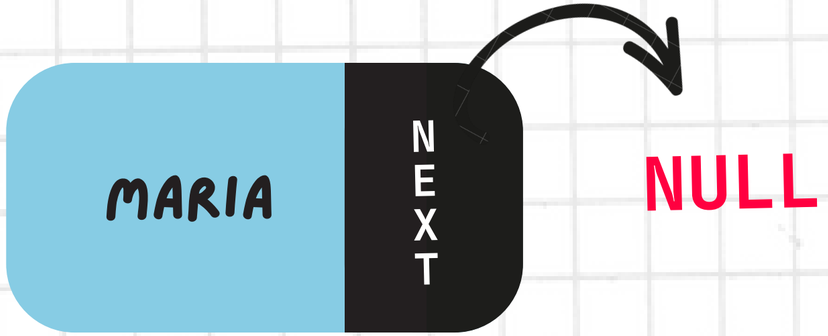
NULL



NULL

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

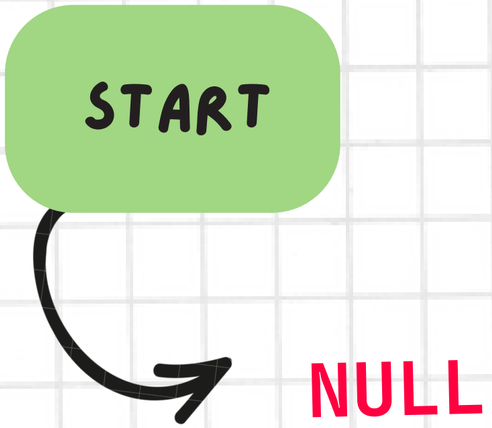


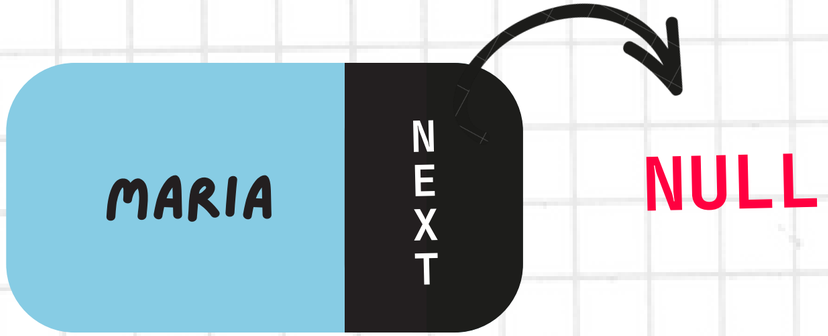



```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

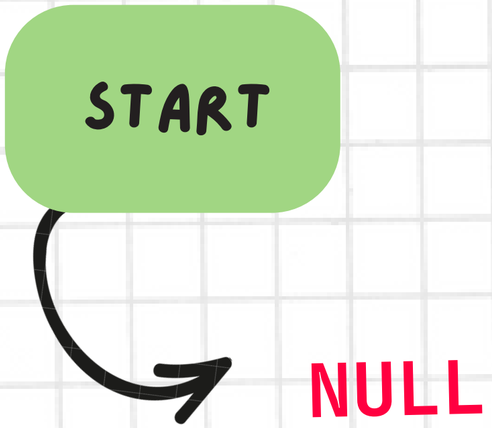
```
THING *addfront(THING *head, THING *newp)
{
    newp -> next = head;
    return newp;
}
THING *addend (THING *head, THING *newp)
{
    THING *p2;
    if (head == NULL)
        return newp;
    for (p2 = head; p2 -> next != NULL; p2 = p2 -> next)
        ;
    p2 -> next = newp;
    return head;
}
THING *addmiddle (THING *head, THING *newp)
{
    BOOL found = FALSE;
    THING *p1, *p2;
    if (head == NULL) {
        head = addfront(head, newp);
        return head;
    }
    p2 = p1 = head ;
    while (!found) {
if (found = strcmp(p1 -> item, newp -> item) >= 1) {
    if (p1 == head) {
        head = addfront(head, newp);
        return(head);
    }
    else {
        p2 -> next = newp;;
        newp -> next = p1;
        return head;
    }
}
    if  (p1 -> next == NULL) {
        head = addend(head, newp);
        return (head);
    }
    p2 = p1; p1 = p1 -> next;

    }
```

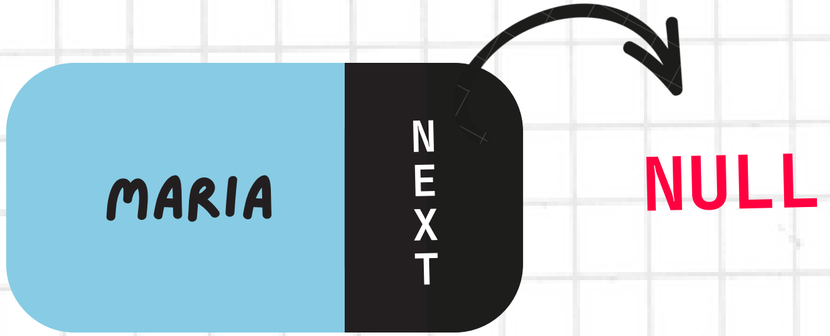





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

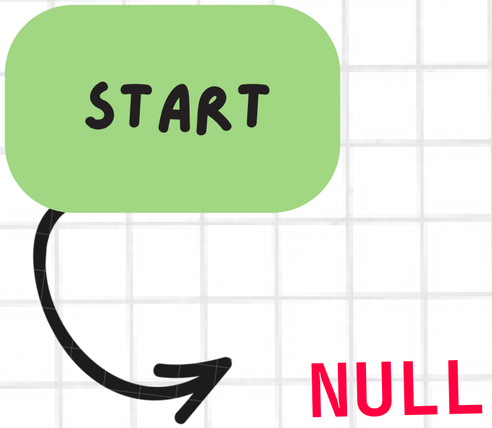
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

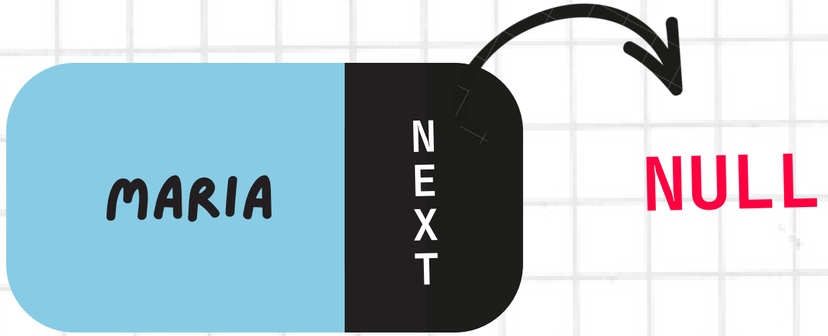





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

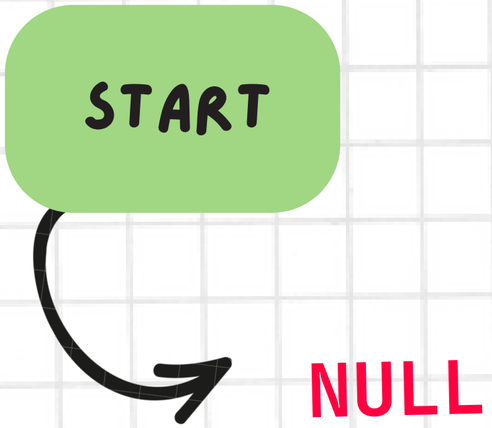
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

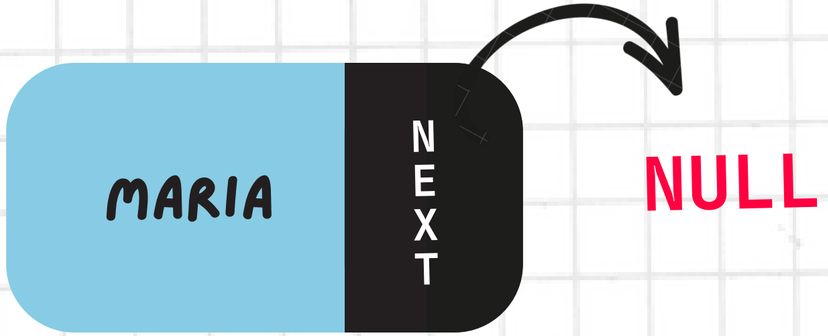





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

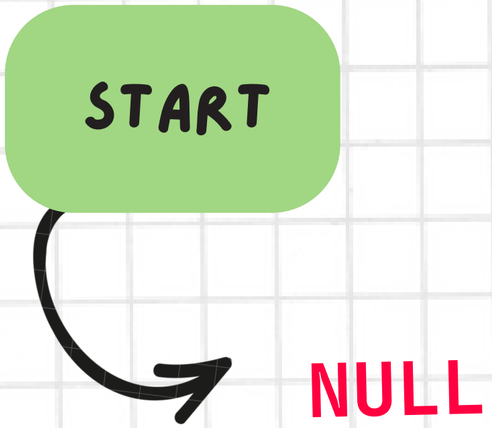
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

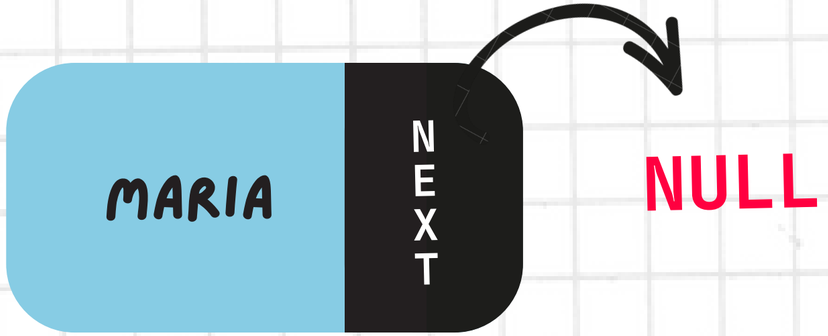





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

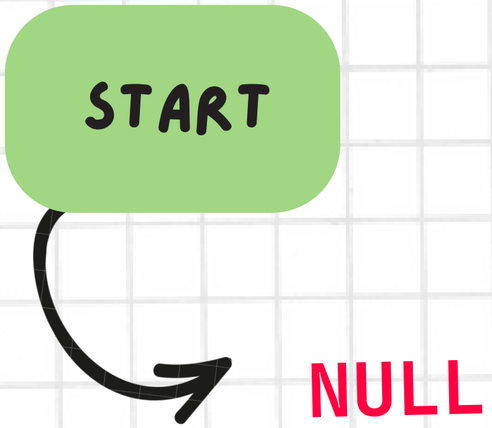
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

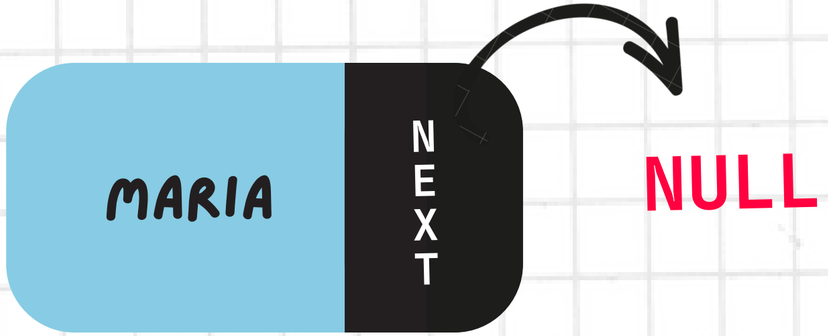





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

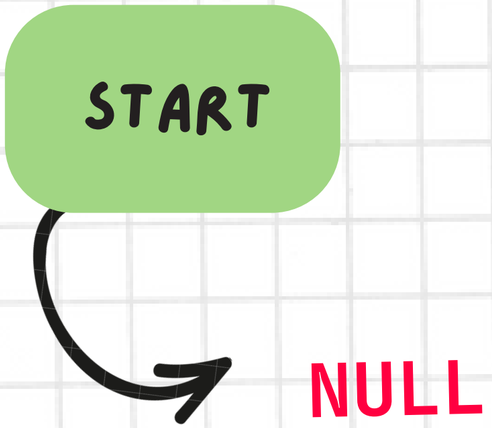
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

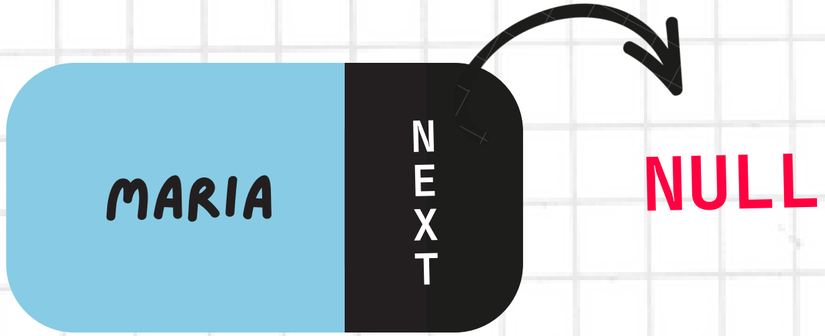





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

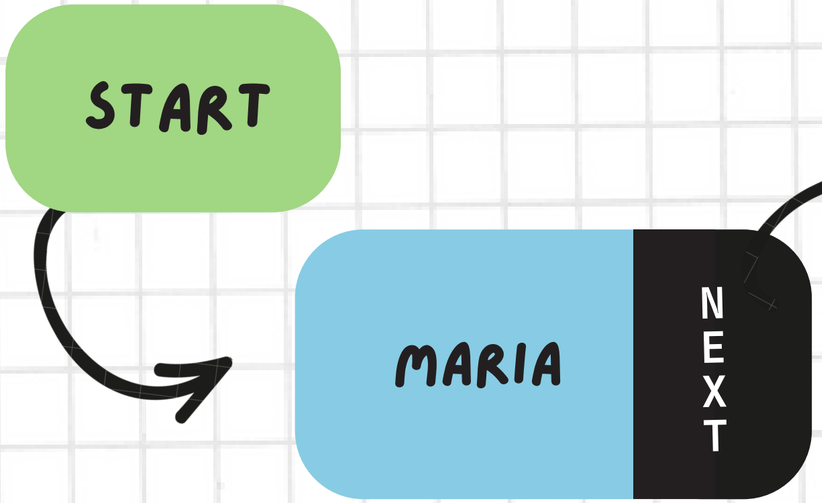
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```





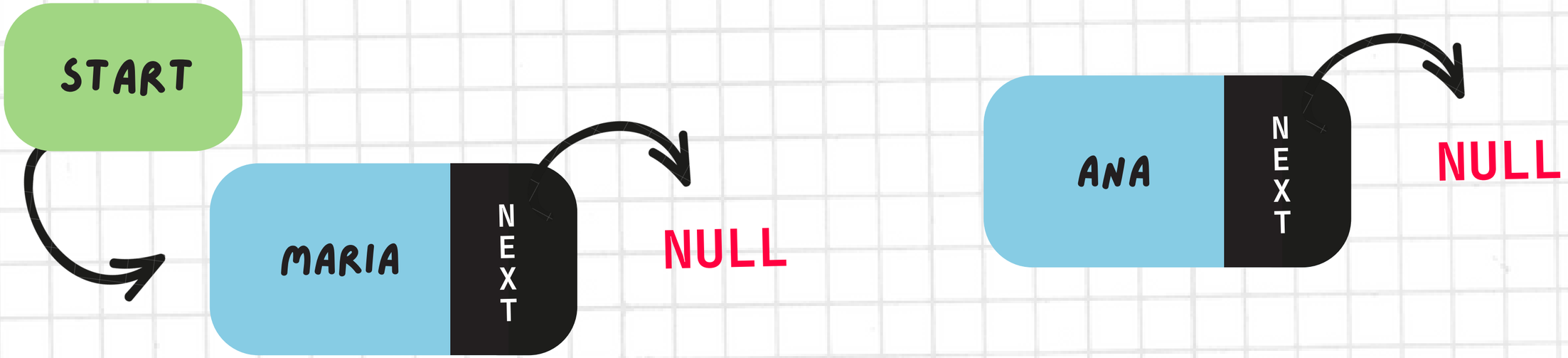

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```



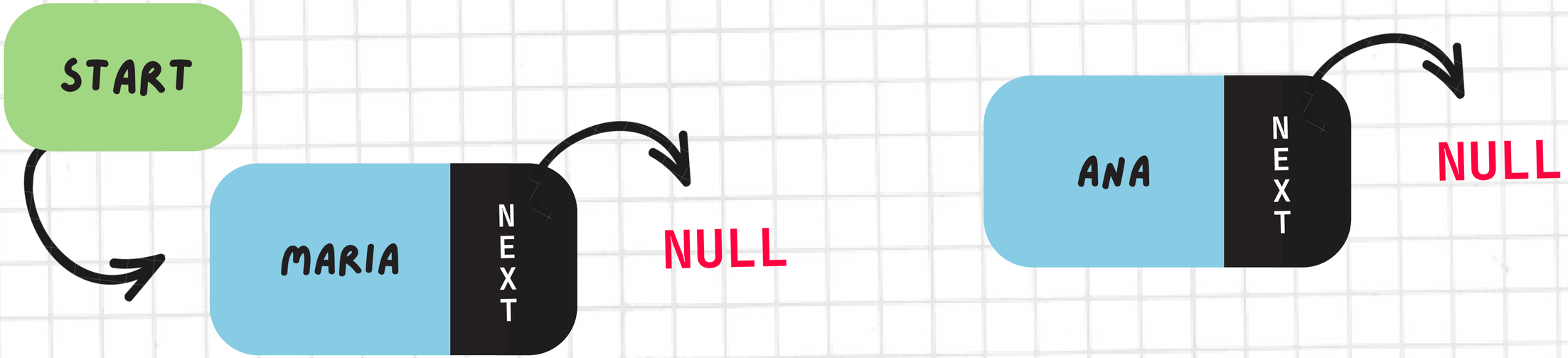

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```



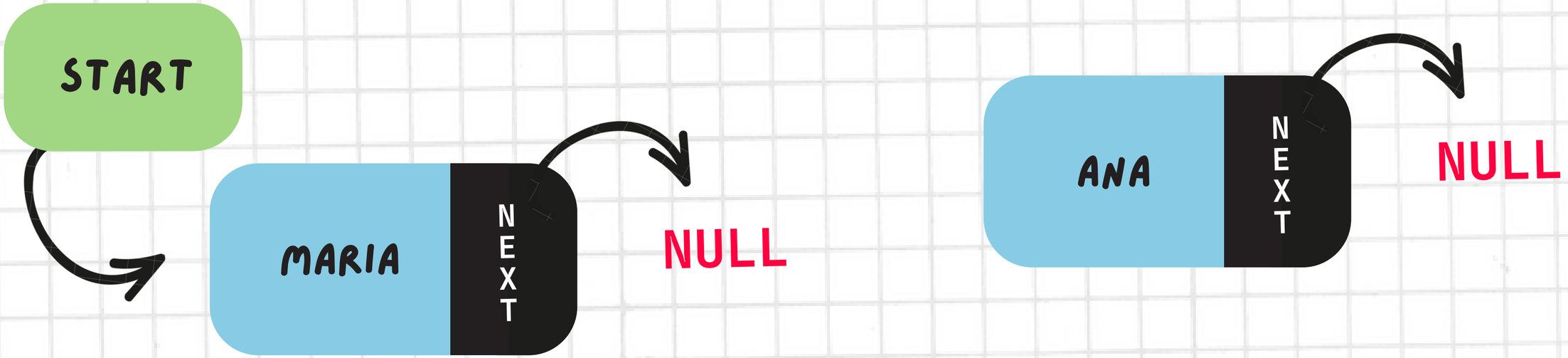

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```



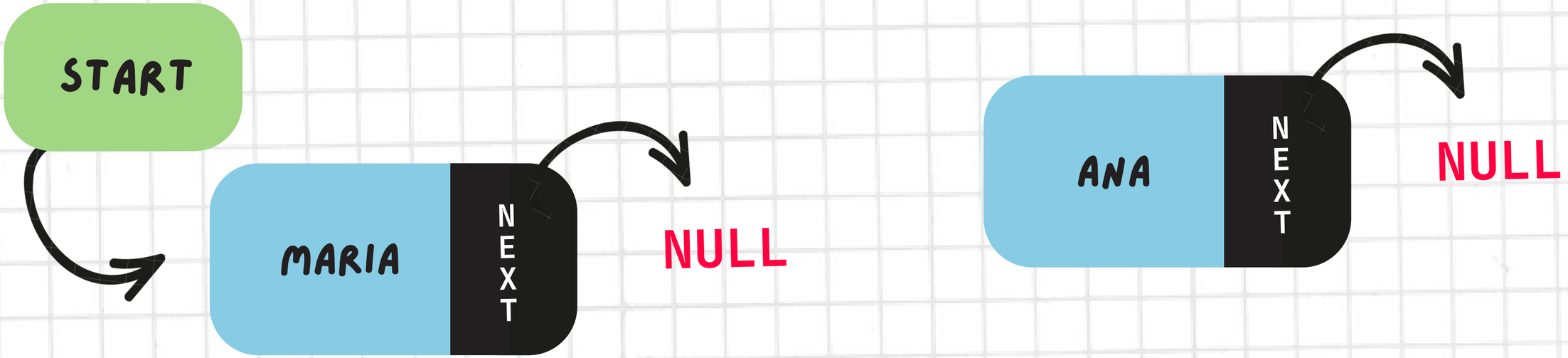

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```



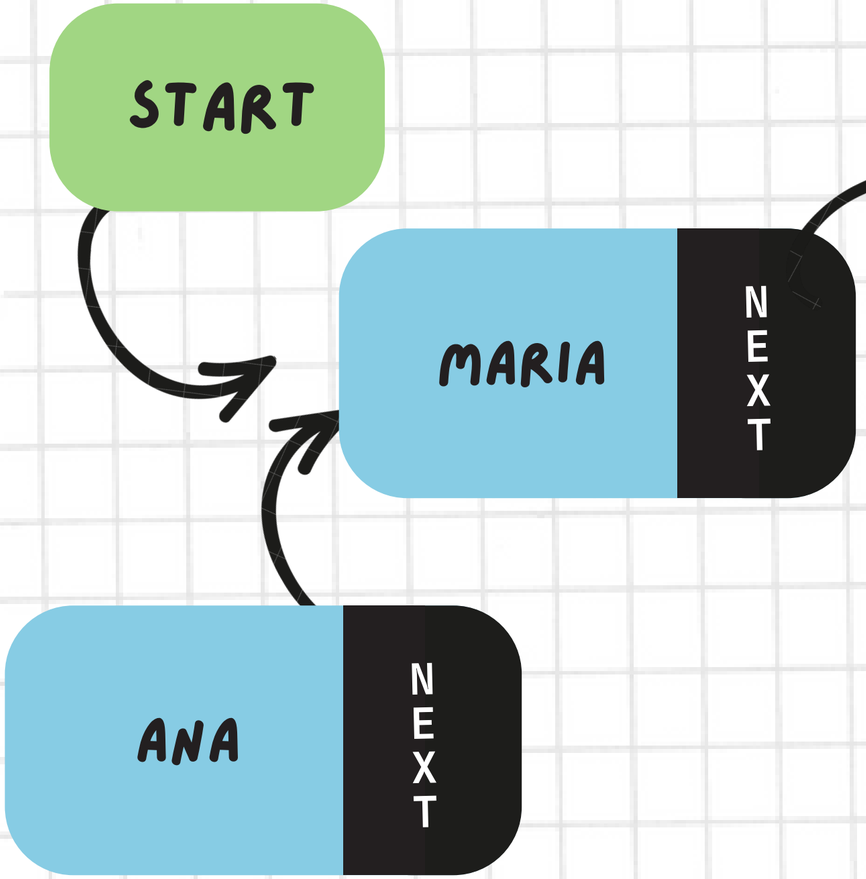

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```



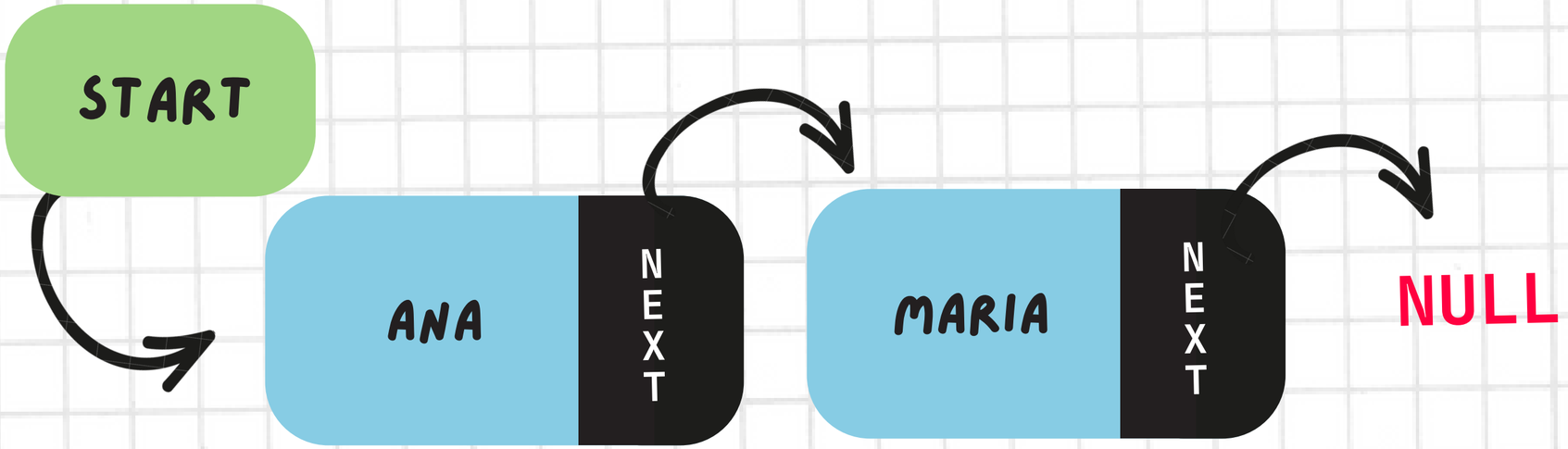

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

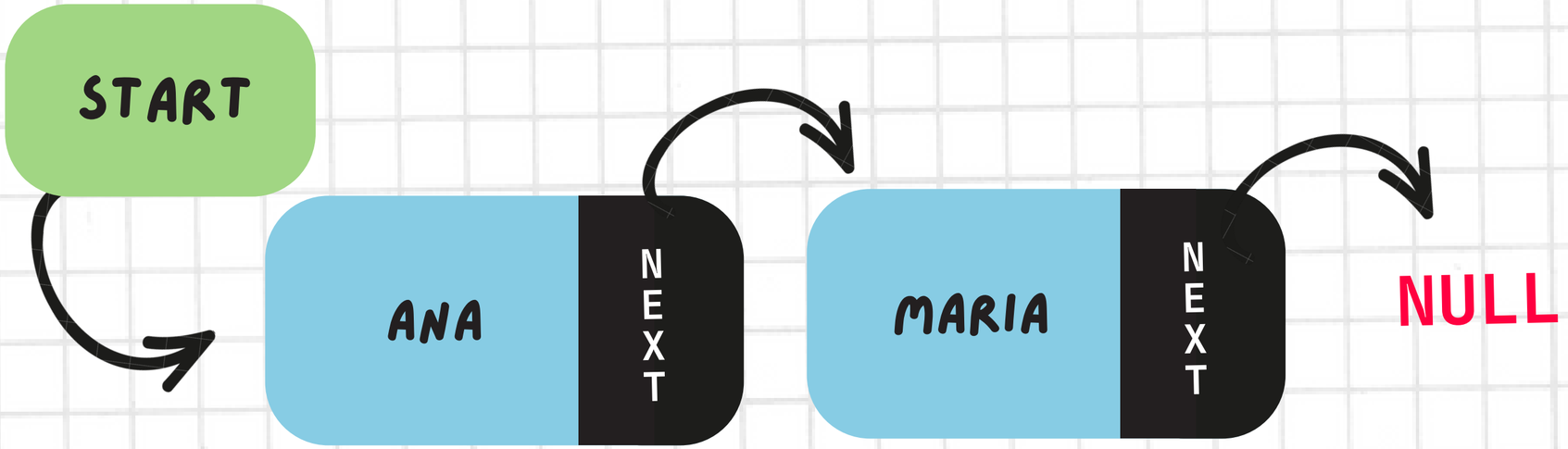



```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

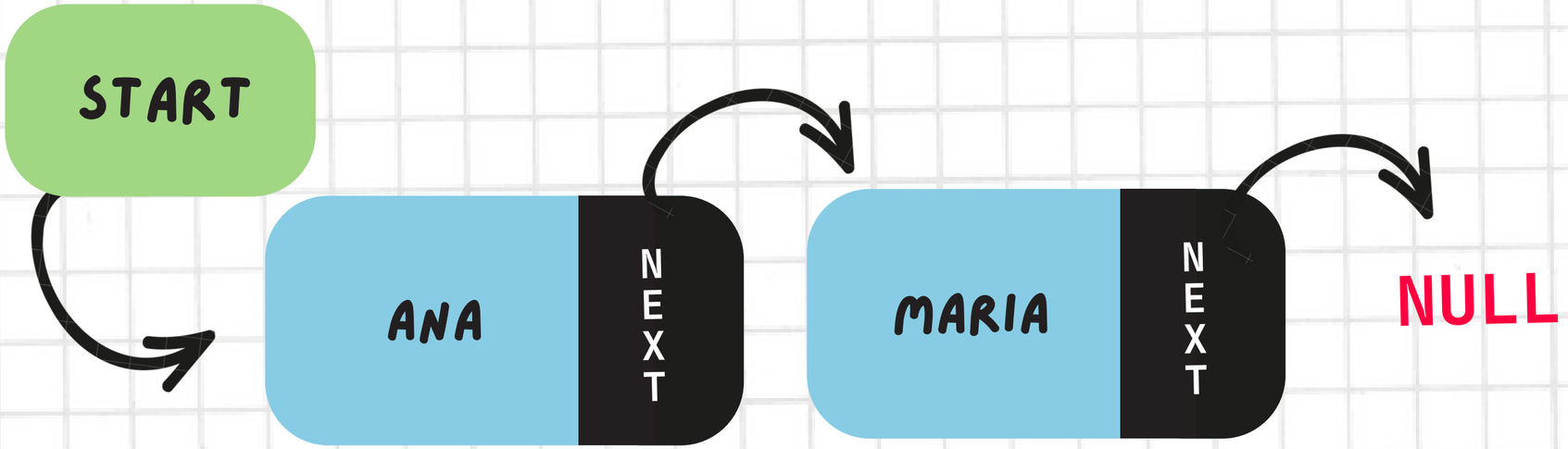
```
INSERTPESSOA(&START, NEWELEMENT("MARIA"));
```

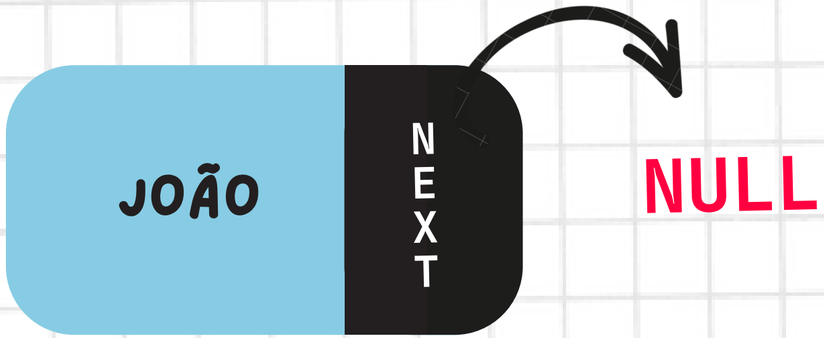



```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```



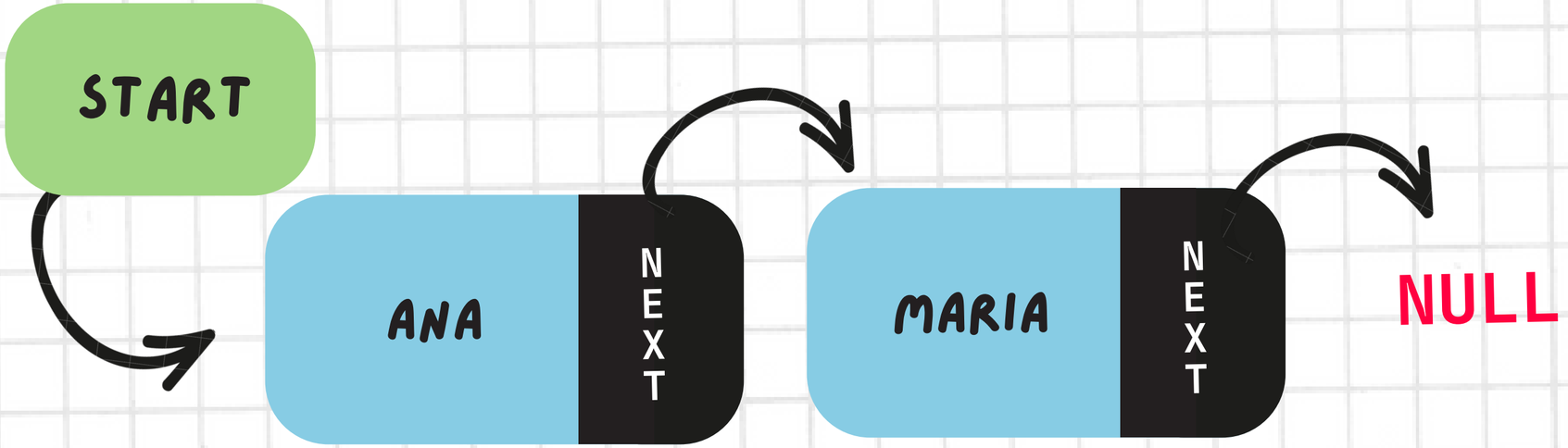

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

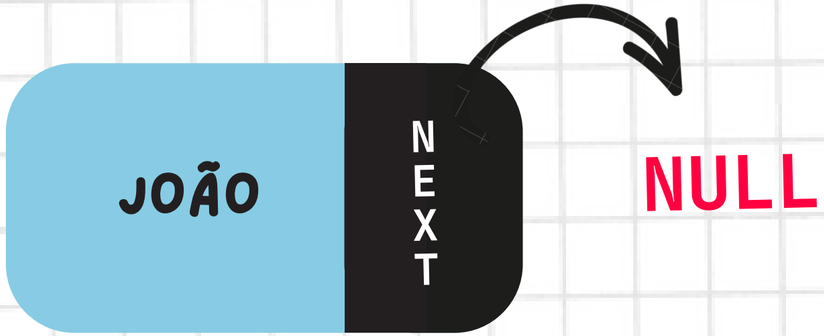





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

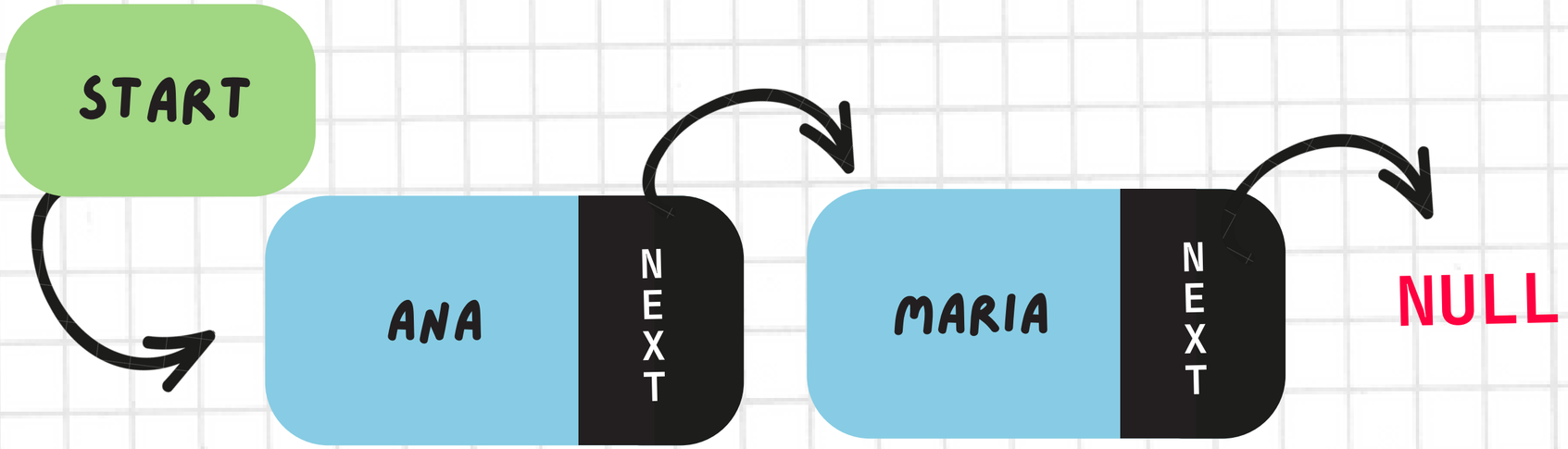
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```

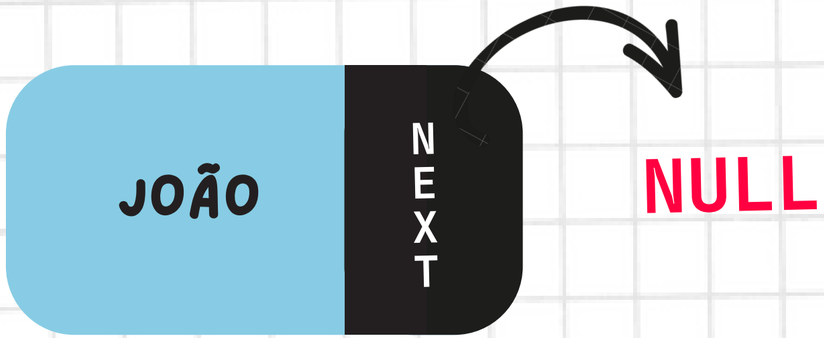





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

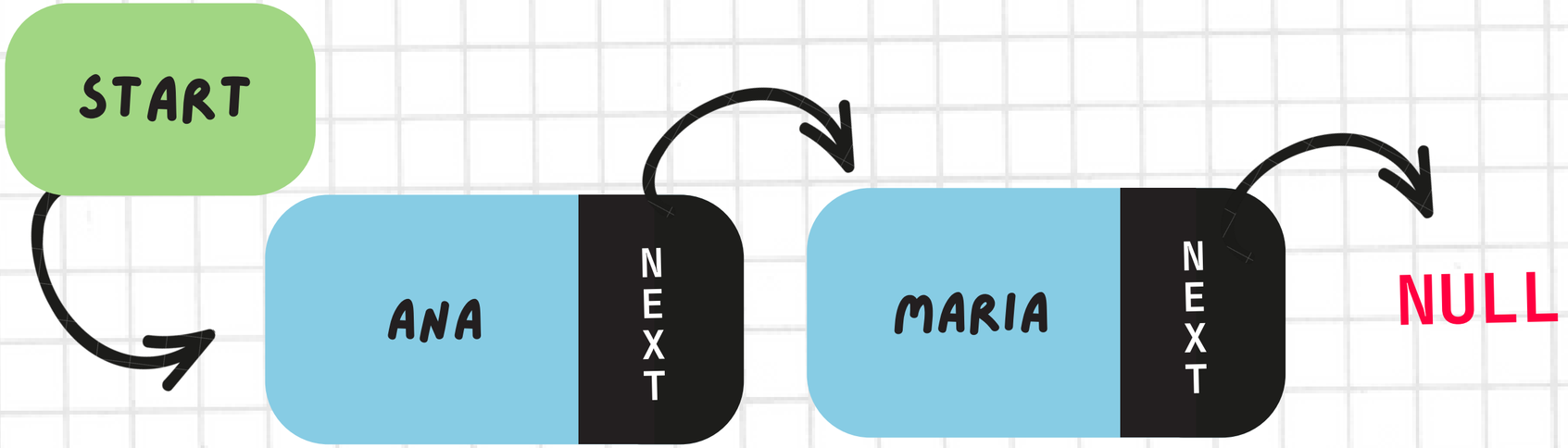
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```

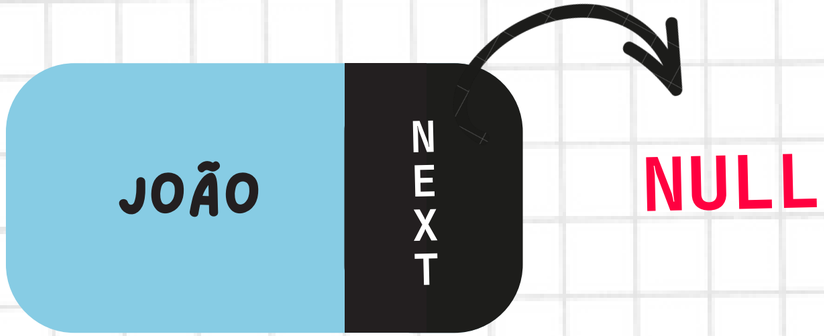





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

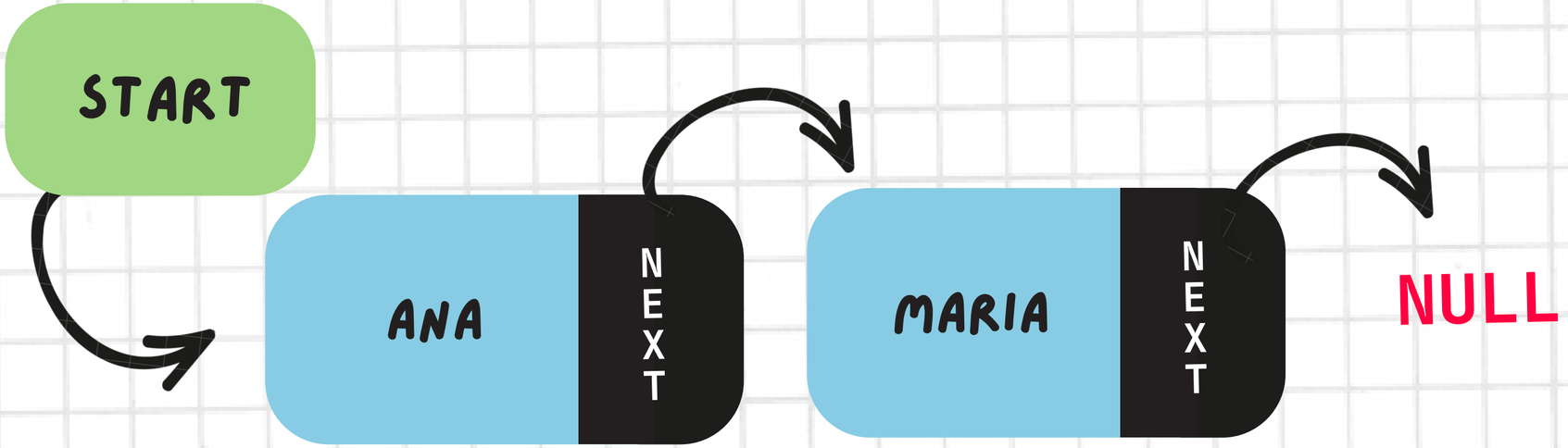
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```

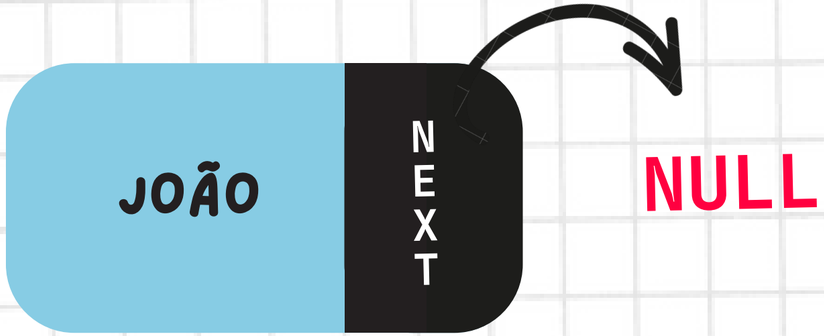





```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

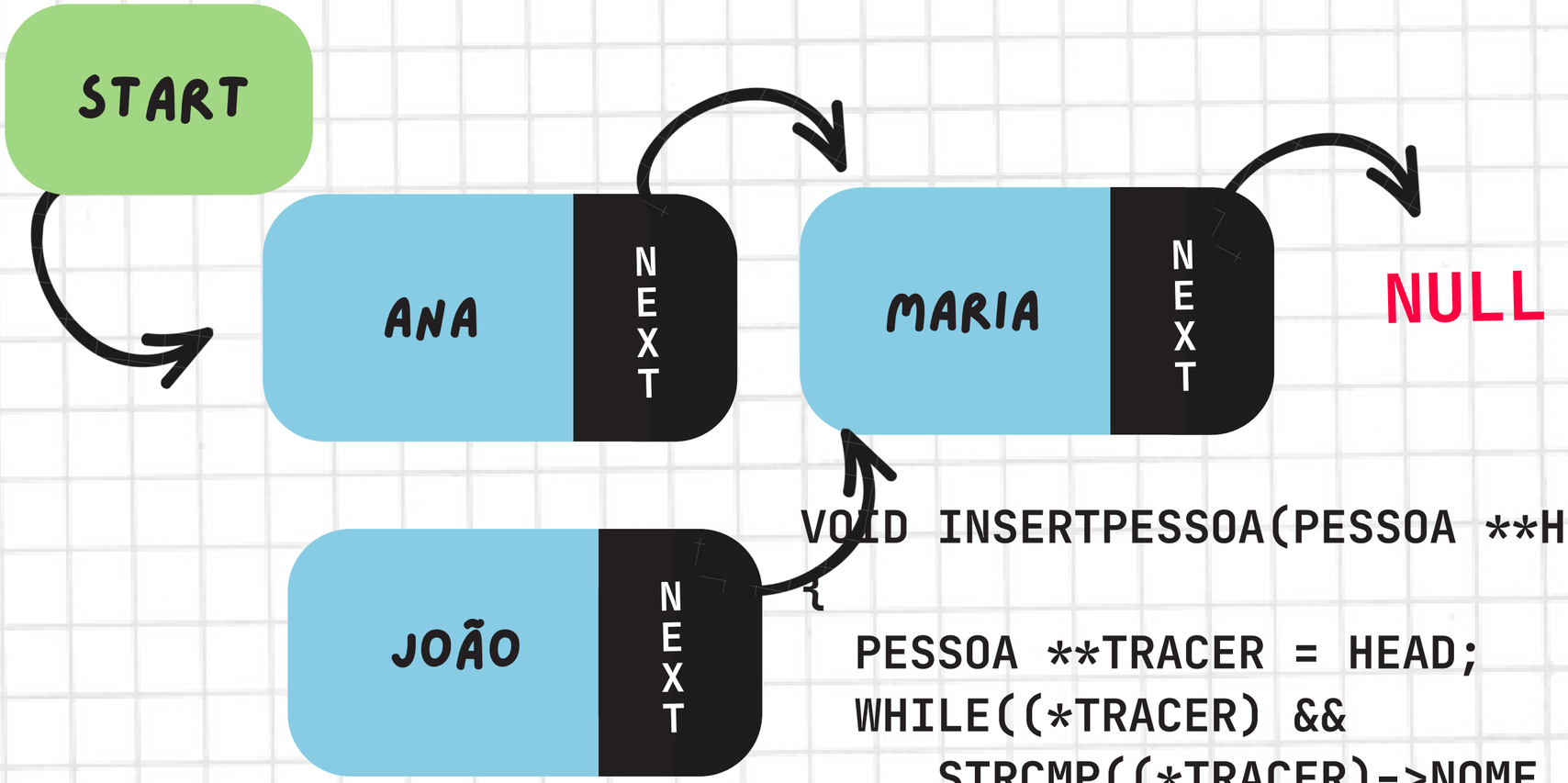
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```





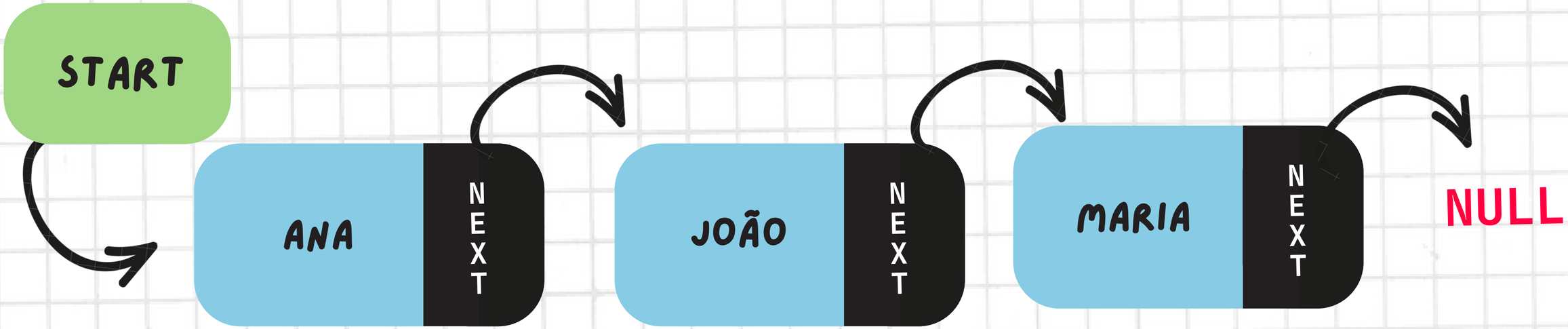

```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
  PESSOA **TRACER = HEAD;
  WHILE((*TRACER) &&
     STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
  {
    TRACER = &(*TRACER)->NEXT;
  }
  NEWP -> NEXT = *TRACER;
  *TRACER = NEWP;
}
```

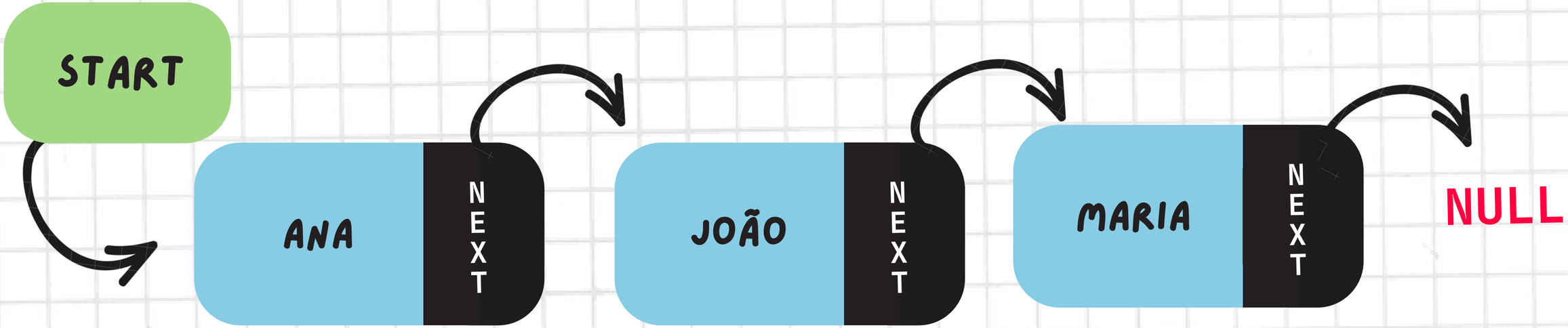
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```




```
VOID INSERTPESSOA(PESSOA **HEAD, PESSOA *NEWP)
{
   PESSOA **TRACER = HEAD;
   WHILE((*TRACER) &&
      STRCMP((*TRACER)->NOME, NEWP -> NOME) < 1)
   {
      TRACER = &(*TRACER)->NEXT;
   }
   NEWP -> NEXT = *TRACER;
   *TRACER = NEWP;
}
```

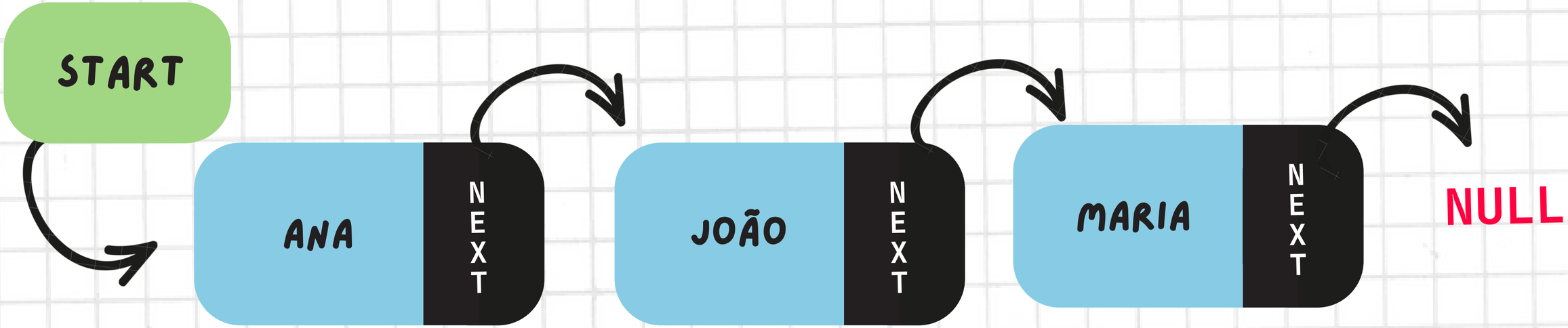
```
INSERTPESSOA(&START, NEWELEMENT("JOÃO"));
```


START
ANA
NEXT
JOÃO
NEXT
MARIA
NEXT
NULL



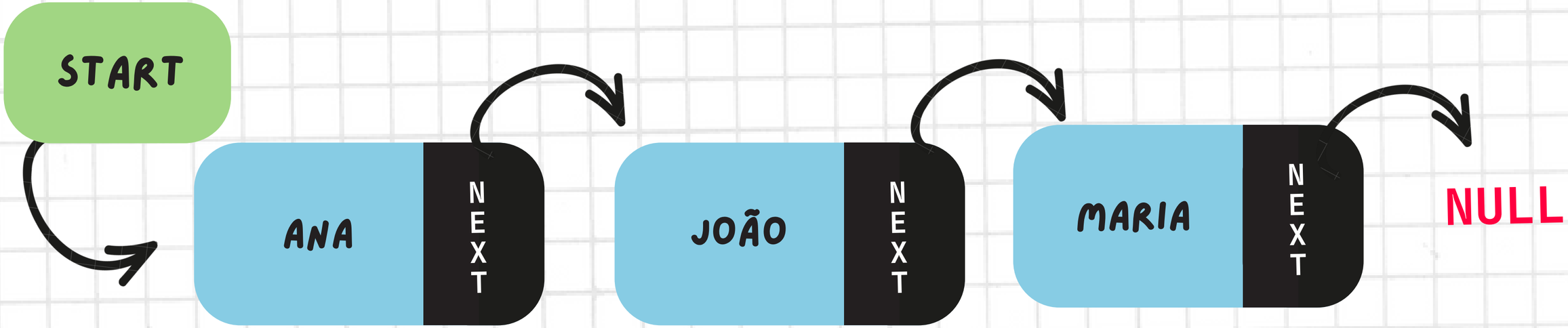
```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

```
PRINTLIST (&START);
```



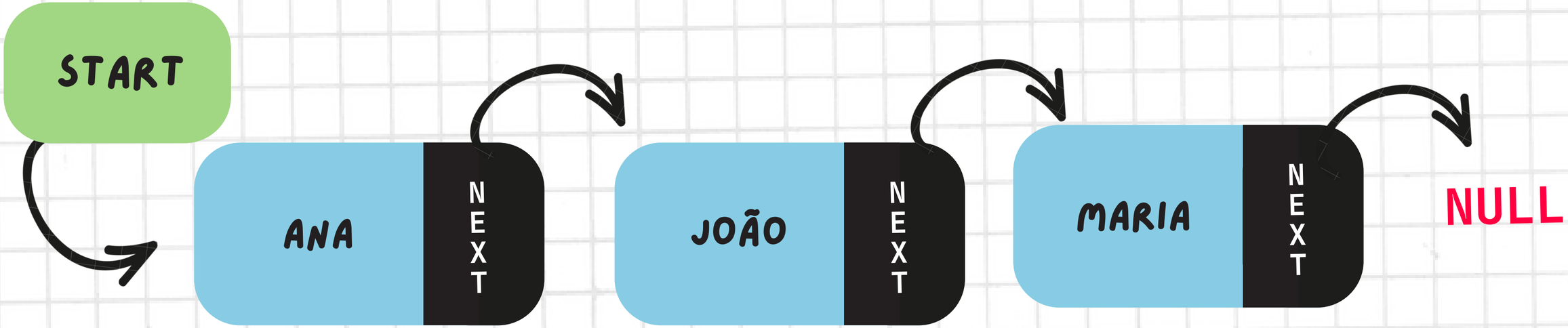
```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

```
PRINTLIST (&START);
```



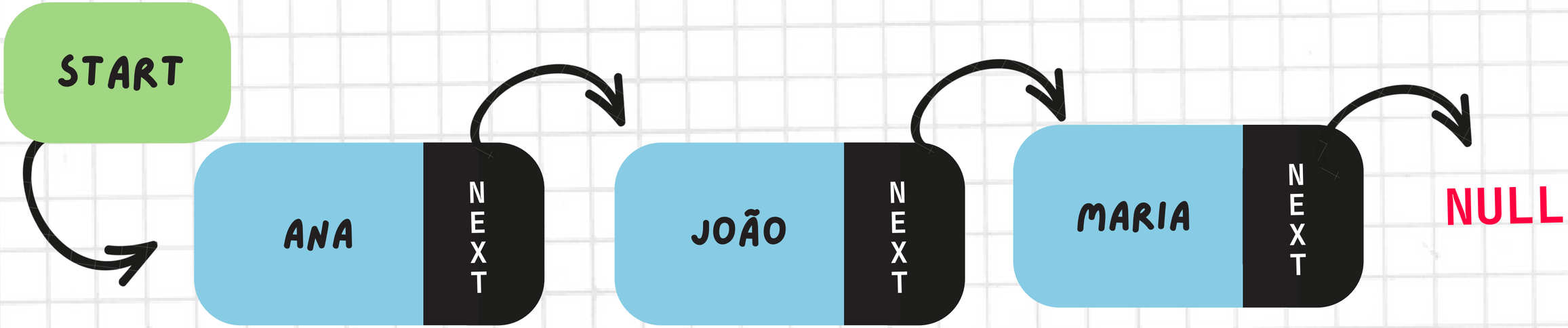
```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

```
PRINTLIST (&START);
```



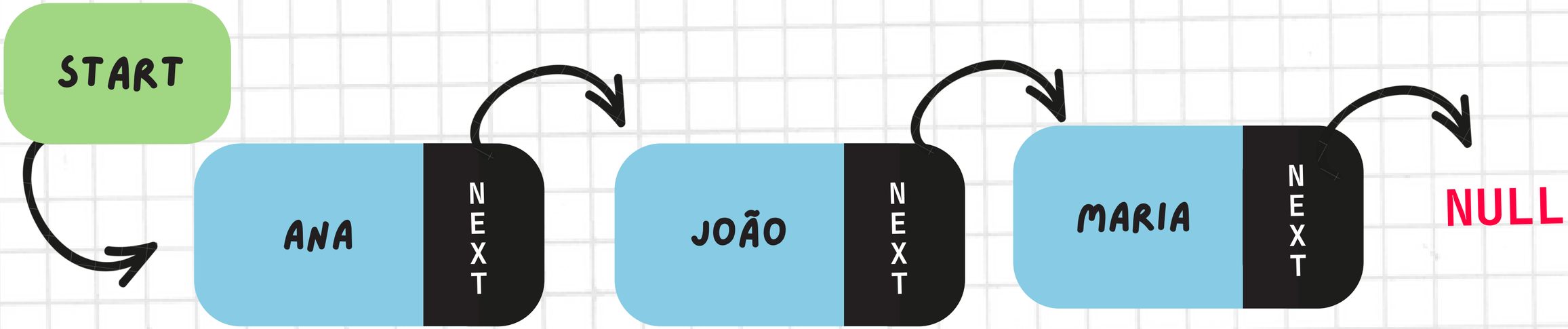
```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

```
PRINTLIST (&START);
```



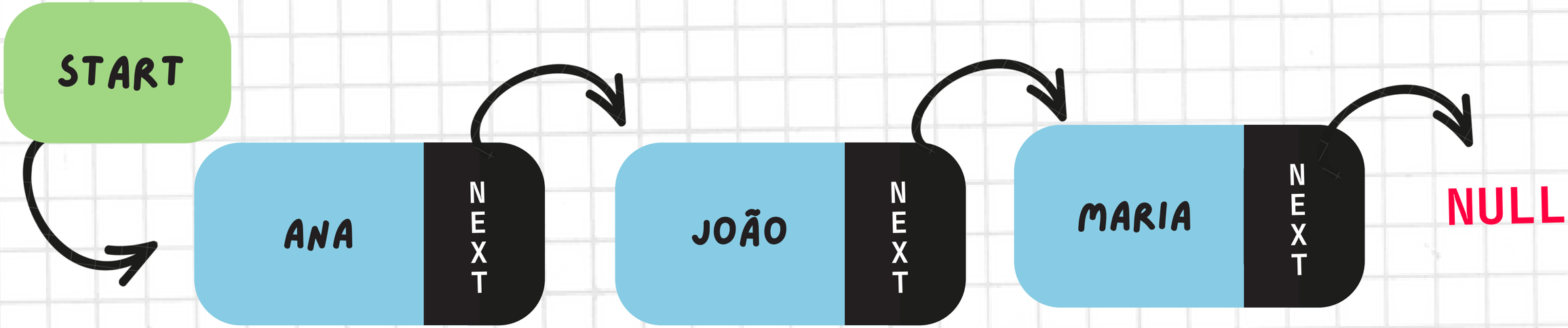
```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

```
PRINTLIST (&START);
```



```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

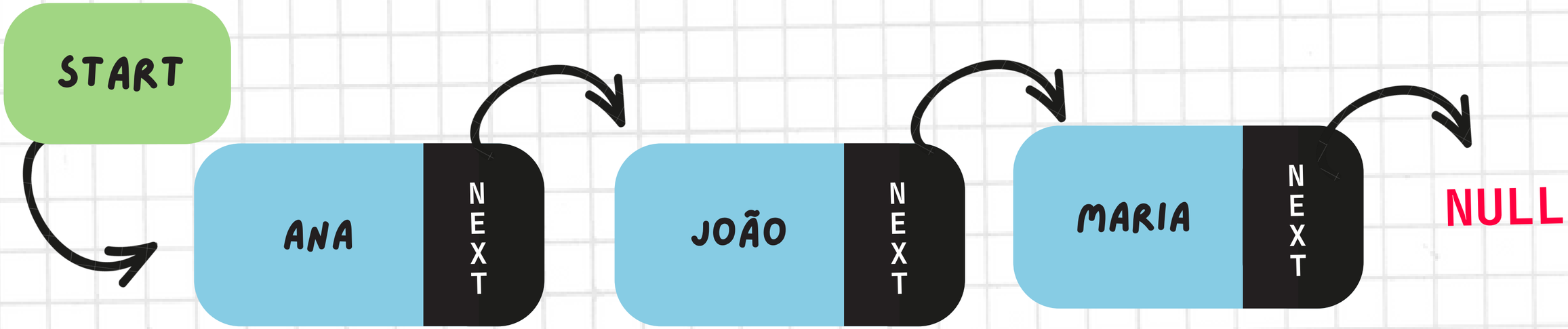
```
PRINTLIST (&START);
```

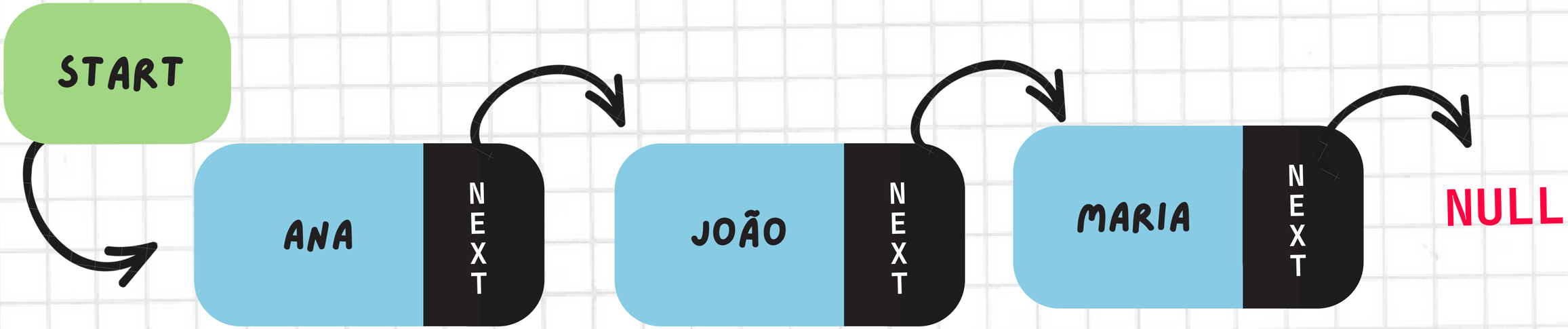



```
ANA
JOÃO
MARIA
```

```
VOID PRINTLIST(PESSOA **HEAD)
{
   PESSOA **TRACER = HEAD;
   WHILE (*TRACER) {
      PRINTF("%S \N",(*TRACER)->NOME);
      TRACER = &(*TRACER)->NEXT;
   }
}
```

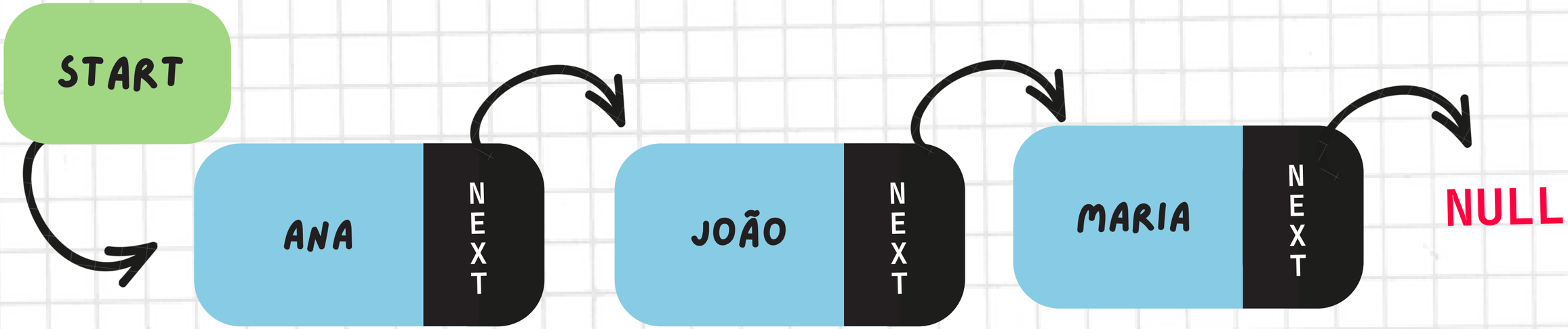
```
PRINTLIST (&START);
```


START
ANA
NEXT
JOÃO
NEXT
MARIA
NEXT
NULL



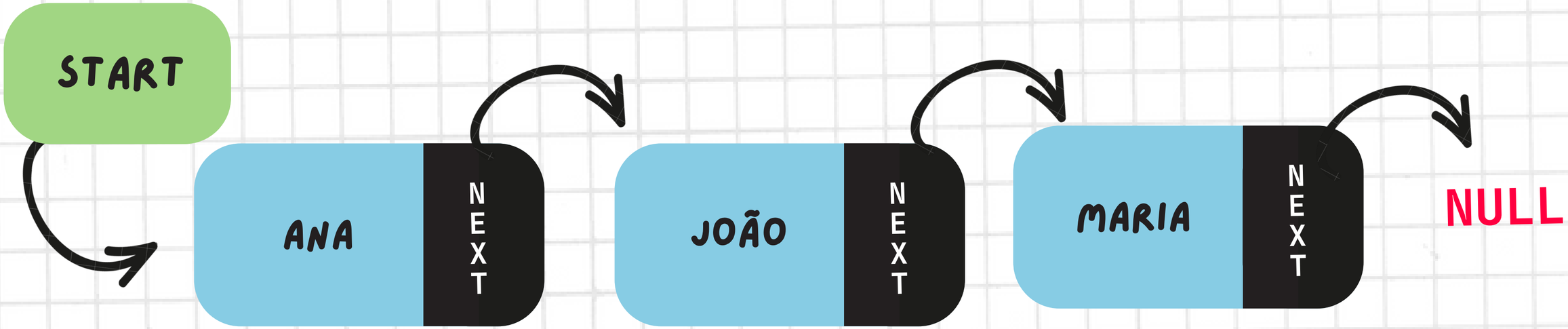

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



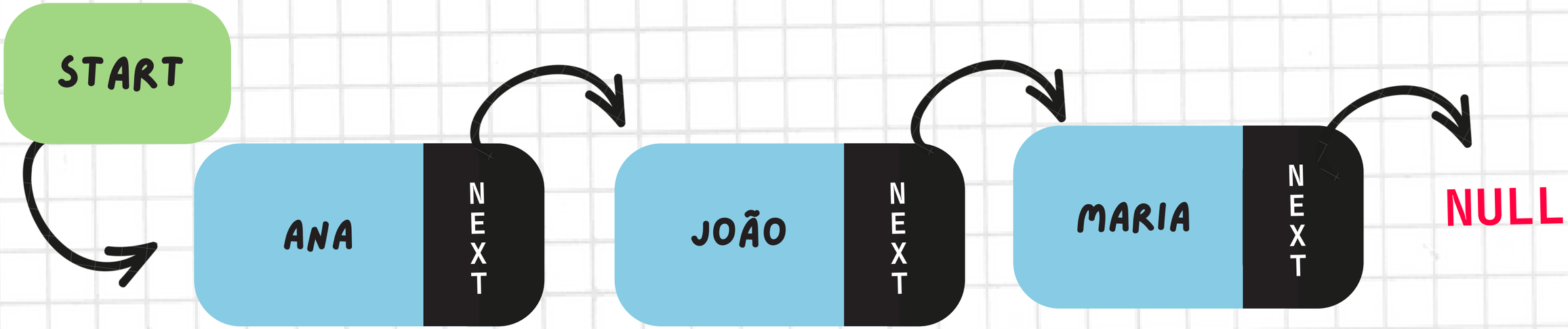

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



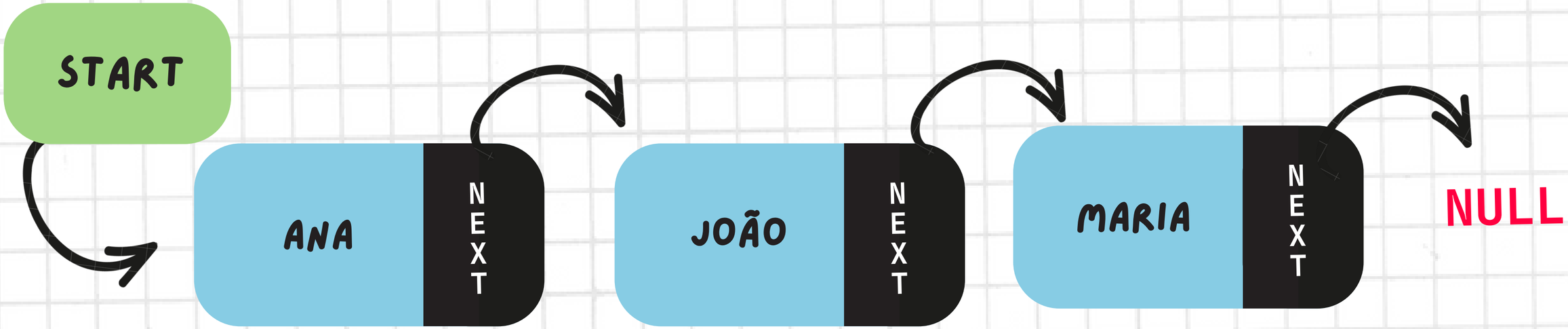

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



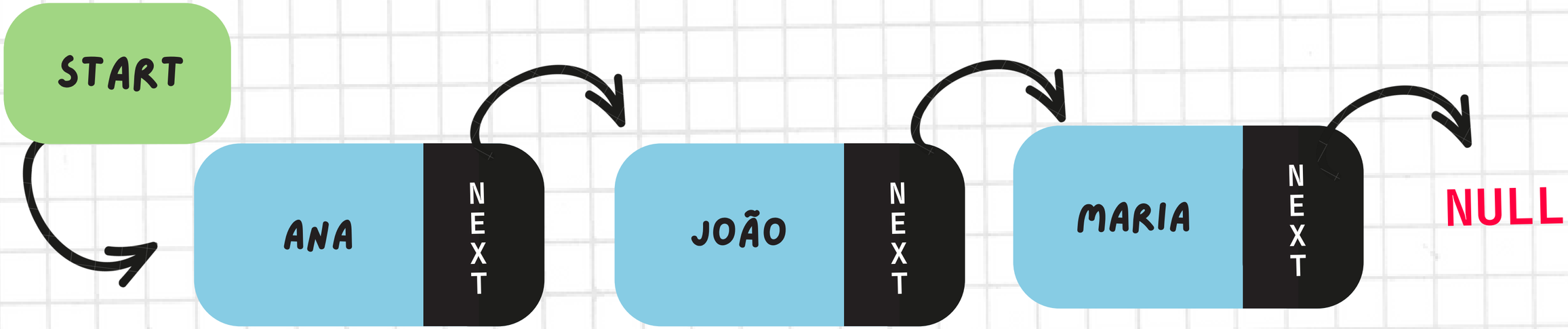

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



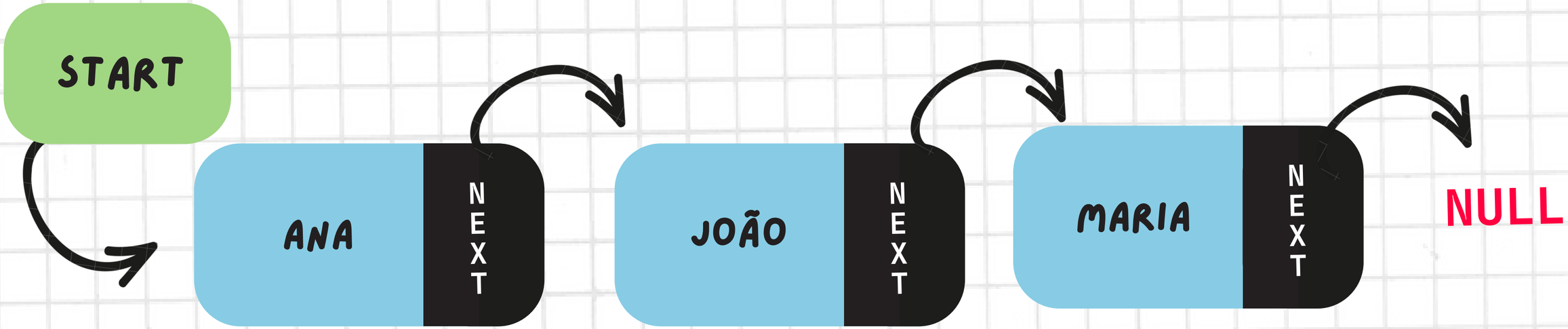

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



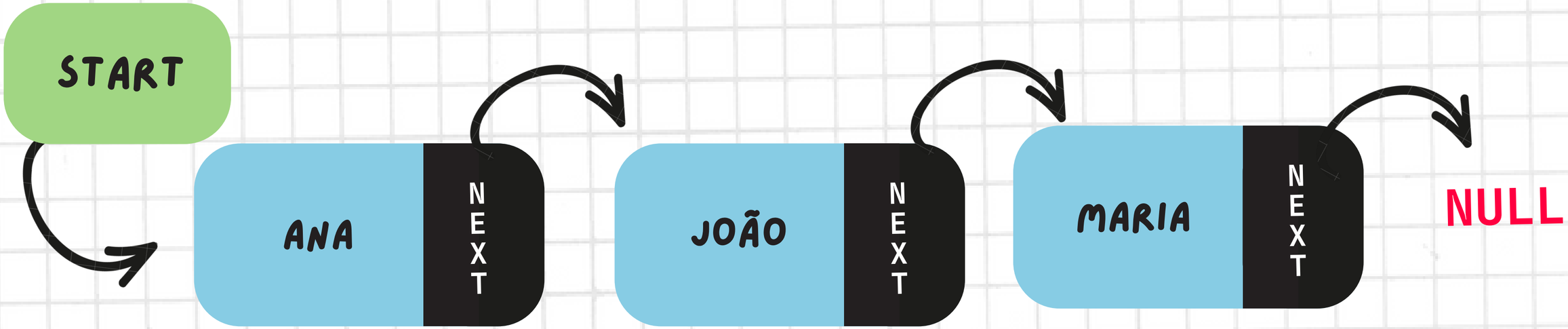

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



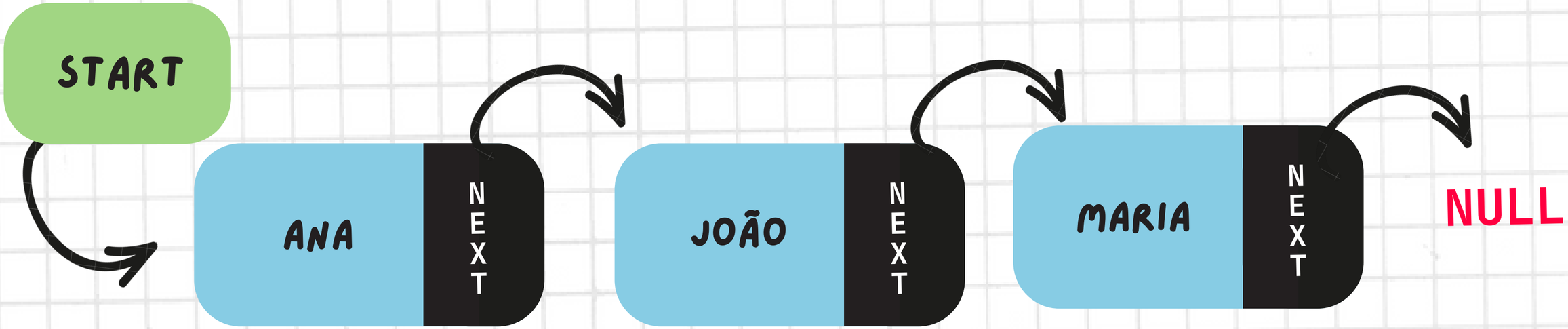

```
VOID REMOVEPESSOA(PESSOA **HEAD, CHAR *TEXT)
{
   BOOL PRESENT = FALSE;
   PESSOA *OLD;
   PESSOA **TRACER = HEAD;

   WHILE((*TRACER) && !(PRESENT = (STRCMP(TEXT,(*TRACER)->NOME) == 0 ) ))
      TRACER = &(*TRACER)->NEXT;
```

```
REMOVEPESSOA(&START, "JOÃO");
```



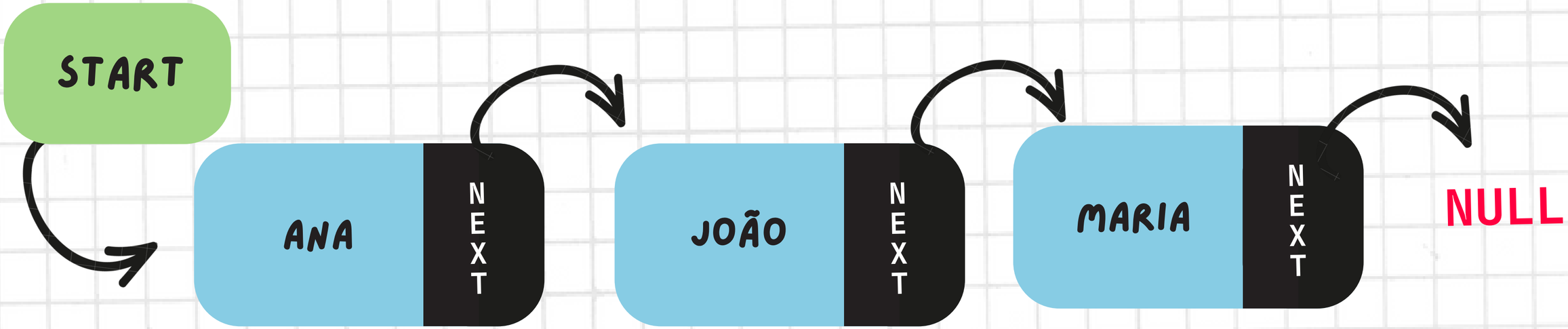

```
IF(PRESENT)
  {
    OLD = *TRACER;
    *TRACER = (*TRACER)->NEXT;
    FREE(OLD -> NOME);
    FREE(OLD);
  }
```

```
REMOVEPESSOA(&START, "JOÃO");
```



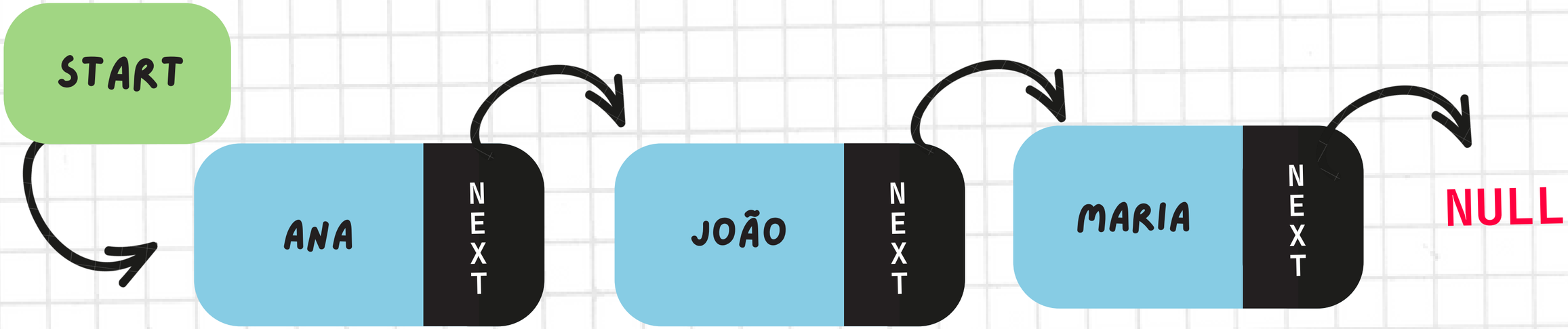

```
IF(PRESENT)
  {
    OLD = *TRACER;
    *TRACER = (*TRACER)->NEXT;
    FREE(OLD -> NOME);
    FREE(OLD);
  }
```

```
REMOVEPESSOA(&START, "JOÃO");
```



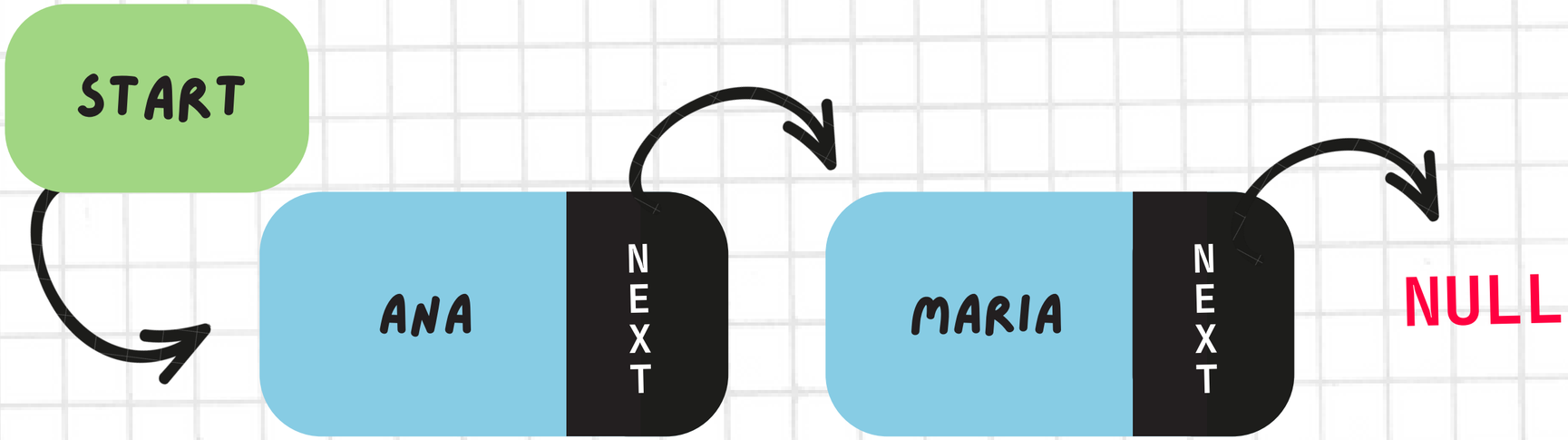

```
IF(PRESENT)
  {
    OLD = *TRACER;
    *TRACER = (*TRACER)->NEXT;
    FREE(OLD -> NOME);
    FREE(OLD);
  }
```

```
REMOVEPESSOA(&START, "JOÃO");
```




```
IF(PRESENT)
  {
    OLD = *TRACER;
    *TRACER = (*TRACER)->NEXT;
    FREE(OLD -> NOME);
    FREE(OLD);
  }
```

```
REMOVEPESSOA(&START, "JOÃO");
```




```
IF(PRESENT)
  {
    OLD = *TRACER;
    *TRACER = (*TRACER)->NEXT;
    FREE(OLD -> NOME);
    FREE(OLD);
  }
```

```
REMOVEPESSOA(&START, "JOÃO");
```

OBRIGADA PELA ATENÇÃO