1. Insert a node at a specific position in a linked list

Complete the C program by writing ins_at_pos() method by considering following criteria:

1. **Insert at Beginning**: If pos == 1, we insert the new node at the start of the list and adjust the start pointer.
2. **Insert at Any Position**:
   - For positions within the list, the while loop traverses to the node just before the desired position.
   - Adjust the next pointers to insert the new node.
3. **Invalid Position Handling**:
   - The function now correctly handles invalid positions by checking if pos is less than 1 or greater than nodectr + 1.
   - If the position is invalid, the new node is freed to prevent memory leaks.

Program:

```
#include <stdio.h>
#include <stdlib.h>

struct slinklist {
   int data;
   struct slinklist *next;
};

typedef struct slinklist node;

node *start = NULL;

int menu() {
   int ch;
   printf("\n 1.Create a list ");
   printf("\n--------------------------");
   printf("\n 2.Insert a node at specified position");
   printf("\n--------------------------");
   printf("\n 3.Display");
   printf("\n--------------------------");
   printf("\n 4. Exit ");
   printf("\n\n Enter your choice: ");
   scanf("%d", &ch);
   return ch;
}

node* getnode() {
   node *newnode;
   newnode = (node *)malloc(sizeof(node));
   printf("\n Enter data: ");
   scanf("%d", &newnode->data);
   newnode->next = NULL;
   return newnode;
}
```

```
void createlist(int n) {
   int i;
   node *newnode;
   node *temp;
   for (i = 0; i < n; i++) {
      newnode = getnode();
      if (start == NULL) {
         start = newnode;
      } else {
         temp = start;
         while (temp->next != NULL)
            temp = temp->next;
         temp->next = newnode;
      }
   }
}

int countnode(node *ptr) {
   int count = 0;
   while (ptr != NULL) {
      count++;
      ptr = ptr->next;
   }
   return count;
}

void display() {
   node *temp;
   temp = start;
   printf("\n The contents of List (Left to Right): \n");
   if (start == NULL) {
      printf("\n Empty List");
      return;
   } else {
      while (temp != NULL) {
         printf("%d-->", temp->data);
         temp = temp->next;
      }
   }
   printf(" X ");
}

void insert_at_pos() {
   node *newnode, *temp, *prev;
   int pos, nodectr, ctr = 1;
   newnode = getnode();
   printf("\n Enter the position: ");
   scanf("%d", &pos);
```

*Write your code here

```
}

void main(void) {
   int ch, n;
   while (1) {
      ch = menu();
      switch (ch) {
         case 1:
            if (start == NULL) {
               printf("\n Number of nodes you want to create: ");
               scanf("%d", &n);
               createlist(n);
               printf("\n List created..");
            } else
               printf("\n List is already created..");
            break;
         case 2:
            insert_at_pos();
            break;
         case 3:
            display();
            break;
         default:
            exit(0);
      }
   }
}
```