**CÁC BẢN CÀI ĐẶT LIST, STACK, QUEUE**

1. **List:**

|  |
| --- |
| typedef struct{  int danhsach[100], size;  }List;  void make\_null\_L(List \*L){   1. >size = 0;   }  void push(List \*L, int u){  L->danhsach[L->size++] = u;  }  int element\_at(List \*L, int i){  return L->danhsach[i - 1];  }  int empty\_L(List \*L){  return L->size == 0;  }  void copy\_list(List \*L1, List \*L2){ //Copy từ ds L2 qua L1  make\_null\_L(L1); // Kh có dòng này ra màn hình đen  for(int u = 1; u <= L2->size; u++)  push(L1, element\_at(L2, u);  } |

1. **Stack:**

|  |
| --- |
| typedef struct{  int nganxep[MAX\_N];  int top\_idx;  }Stack;  void make\_null(Stack \*S){  S->top\_idx = 0;  }  void push\_back(Stack \*S, int u){  S->nganxep[S->top\_idx++] = u;  }  int top(Stack \*S){  return S->nganxep[S->top\_idx - 1];  }  void pop(Stack \*S){  S->top\_idx--;  }  int empty(Stack \*S){  return S->top\_idx == 0;  } |

1. **Queue:**

|  |
| --- |
| #define MAX\_SIZE 50  typedef struct{  int hangdoi[MAX\_SIZE];  int front, rear;  }Queue;  void make\_null(Queue \*Q){  Q->front = 0;  Q->rear = -1;  }  void enQueue(Queue \*Q, int u){  Q->rear++;  Q->hangdoi[Q->rear] = u;  }  int front(Queue \*Q){  return Q->hangdoi[Q->front];  }  void deQueue(Queue \*Q){  Q->front++;  }  int empty(Queue \*Q){  return Q->front > Q->rear;  } |